UNIVERSITÉ DE SHERBROOKE
Faculté de génie
Département de génie électrique et de génie informatique

CONTRÔLE EN FORCE SÉCURITAIRE D'UNE PLATEFORME OMNIDIRECTIONNELLE NON-HOLONOME

SECURED FORCE GUIDANCE OF AN OMNIDIRECTIONAL NON-HOLONOMIC PLATFORM

Mémoire de maîtrise
Specialité : génie électrique

Julien FRÉMY

Jury: François MICHAUD (directeur)
Michel LAURIA (co-directeur)
Roch LEFEBVRE (rapporteur)
Yves BÉRUBÉ-LAUZIÈRE (examinateur)

Sherbrooke (Québec) Canada Janvier 2011

To my family, without you I'd be nothing.

RÉSUMÉ

Pour qu'un jour les robots puissent cohabiter avec les humains, ils doivent pouvoir interagir physiquement avec leur environnement, comme par exemple la capacité de réagir aux poussées. Or pour le moment, très peu de recherches ont été portées sur de telles interactions physiques. De plus, celles-ci aboutissent souvent à des prototypes dédiés à cet usage, disposant de poignées équipées de capteurs de force. AZIMUT-3, un robot mobile omnidirectionnel et non-holonome novateur développé à l'IntRoLab (le Laboratoire de robotique intelligente, interactive et interdisciplinaire de l'Université de Sherbrooke), est capable de percevoir des couples extérieurs au niveau de ses roues à partir d'un mécanisme de direction active contrôlable en impédance. Il en résulte un effet de suspension horizontale réalisé grâce à l'impédance contrôlable des moteurs orientant ses roues. Cet effet de souplesse renforce la pertinence de l'utilisation de ce robot dans le cadre d'interactions physiques. En réponse à ces interactions, la plateforme doit être en mesure de conjuguer ces forces avec son déplacement sécuritaire, autre élément peu considéré à ce jour avec les robots contrôlé en force. L'objectif est d'assurer la sécurité des déplacements tout en laissant à l'utilisateur un maximum de contrôle sur le robot.

Ce mémoire présente deux contributions importantes, soit : le développement d'un algorithme permettant d'évaluer les efforts appliqués sur AZIMUT-3 à partir des couples perçus au niveau de ses roues ; et l'exploitation de ces efforts avec la détection d'obstacles perçus à partir d'un capteur laser afin de réaliser un contrôle partagé et sécuritaire de la plateforme. Les résultats expérimentaux obtenus à même la plateforme indiquent que le système est fonctionnel, sécuritaire et arrive à des performances comparables à l'utilisation d'un capteur de force à six degrés de liberté, à coût moindre et avec un champ d'interaction plus grand. Le développement réalisé a aussi permis de mettre en place un système original de simulation liant les logiciels Webots et la librairie ROS (*Robot Operating System*) de Willow Garage.

Ce travail constitue donc une première dans l'objectif de pouvoir interagir de façon naturelle avec des robots pour les positionner ou les amener dans des endroits précis, comme ça pourrait être utile avec un chariot d'épicerie ou une marchette robotisée par exemple.

Mots-clés : ROS, simulation dynamique, dynamique du robot, robot mobile, évitement d'obstacle, contrôle en force, contrôle partagé

ABSTRACT

For robots to operate in real life settings, they must be able to physically interact with the environment, and for instance be able to react to force-guidance interactions. However, only a few research projects have addressed such capabilities, developing prototypes that have to be pushed from their handle bars. AZIMUT-3 is a novel omnidirectional non-holonomic mobile robot developed at IntRoLab (Intelligent, Interactive and Interdisciplinary Robot Lab, Université de Sherbrooke) with force-controlled active steering. This results in a horizontal suspension effect for which the mechanical impedance of the steering actuators can be controlled. This makes the platform ideal for developing physical guidance algorithms. One such algorithm is secured shared-control, making the platform go in the direction of the user pushing the robot while still making it move safely by avoiding obstacles. Such capability is somewhat novel in the field, and the objective is to provide safe navigation with maximum control to the user.

This Master's thesis has two important contributions: an algorithm to estimate the applied efforts on AZIMUT-3 from torque measurements on its wheels; an algorithm to use these efforts with obstacle detection using laser range finder data to implement a safe, shared-control approach. Experimental results using the real platform demonstrate feasibility and safe control of the system, with performances similar to using a six degrees of freedom force sensor but at lower cost and with a broader area for shared control. Our implementation also resulted in coupling the simulation environment Webots with the ROS (Robot Operating System) library from Willow Garage, to help develop our approach in simulation before using AZIMUT-3.

Overall, our work is the first in demonstrating how it is possible to naturally interact by physically moving or positioning a mobile platform in real life settings, a capability which could be useful for instance in the design of powered shopping carts or active walkers.

Keywords: ROS, dynamic simulation, robot dynamics, mobile robot, collision avoidance, force-sensitive control method, shared-control

ACKNOWLEDGEMENTS

I want to thank François Michaud for his constant support and encouragements during this Master's thesis. His advices were always useful. I also want to thank Michel Lauria because he made this adventure at the Université de Sherbrooke possible and worthwhile.

I also want to acknowledge the help brought by the members of IntRoLab, and in particular François Ferland and Dominic Letourneau for their availability and efficiency in debugging some of my code in C++. Thanks as well to Marc-Antoine Legault for his help with the design of mechanical parts for my project, and to Benoît Heintz and Lionel Clavien for all the insightful discussions we had.

Finally, thanks to my close relatives and friends, and in particular those who have made the trip to Sherbrooke to visit. Seeing you gave me the strength to overcome the sometimes difficult challenges of completing this Master's thesis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

According to a recent study, the personal service robot domain is an expanding industry which will grow to $19 Billion by 2017 [29]. The robots operating in real-life settings will take various shapes and roles, but they will surely be able to interact with people in natural ways, one of which involves moving the robot to a desired location. To date, joystick, touchscreen, gesture and vocal interfaces have been developed, but they all require a form of training before users can exploit them efficiently. The most natural way to position a robot is to physically guide it to the desired location. This requires the robot to be able to sense forces applied on it to determine where to move.

AZIMUT-3 is an innovative robotic platform that embeds torque sensors on its steering wheels' axis, giving information on the efforts applied from almost any location on the platform. One challenge to force-guide AZIMUT-3 is to derive the overall force and torque applied on the platform using the forces and torques sensed from the four steered wheels. To include safety during motion, other challenges are to exploit such information to derive the user's intent, and to combine it with information about the environment, such as the location of obstacles, to come up with secure and efficient motion assistance capability when physically guiding the robot.

This work is organized as follows. Chapter 2 presents a review of force-guided platforms and their main characteristics in terms of locomotion, interfaces, force sensing, perception, localization, navigation and shared-control, putting into perspective how AZIMUT-3 clearly distinguishes itself from other platforms. Chapter 3 explains the methodology used to overcome the two challenges previously explained. Chapter 4 presents our work in the form of a journal paper, which was submitted to *IEEE Transactions on Robotics*. A brief conclusion is given in Chapter 5.

CHAPTER 2

FORCE-GUIDED ROBOTS

Force-guidance of mobile platforms is a research subject that only a few have addressed so far, with applications such as intelligent walkers and object transportation [13, 36]. Table 2.1 presents the different force-guided robots found in the scientific literature and reviewed in this chapter. Such platforms tackle multiple dimensions associated to robotics, such as locomotion, motion control, perception, navigation, and shared control. AZIMUT-3 is listed in tables throughout Section 2 for comparison, but its characteristics are discussed in Section 3 and 4.

Table 2.1 Existing force-guided robots.

Robot	**Source**	**Refs.**	**Year**
Active RT-Walker	Tohoku University, Japan	[5]	2005
AZIMUT-3	Université de Sherbrooke, Canada	[21]	2006
Care-O-Bot II	Fraunhofer IFA, Germany	[12]	2007
CMU Robotic Walker	Carnegie Mellon University, USA	[25]	2003
COOL-Aide	MARC, University of Virginia, USA	[41]	2008
Gait Assistant	Hanyang University, South Korea	[26]	2004
Guido	DISAM, University of Madrid, Spain	[31]	2005
Locomaid	University of Genova, Italy	[24]	2002
MOBIL	Scuola Superiore Sant'Anna, Pisa, Italy	[32]	2002
Passive RT-Walker	Tohoku University, Japan	[14]	2007
Power-Assisted Walker	Research Laboratory of Hitachi, Japan	[8]	2004
SmartWalker	Massachussets Institute of Technology, USA	[37]	2006
UTS Assistant	University of Technology of Sidney, Australia	[22]	2006
WAR	Inha University, South Korea	[34]	2005

2.1 Locomotion

Table 2.2 lists the locomotion modalities of force-guided robots. Some are manually propelled and cannot move autonomously, but use motors or brakes to orient the device. They are referred to as passive devices. Passive locomotion has several advantages :

- **Security**. Passive robots cannot physically damage their environment, because they cannot move by themselves.
- **Reduced cost**. Passive robots do not embed powerful motors, thus the electromechanics required are more affordable.
- **Simplified interface**. Because there are no propelling motors on a passive robot, the range of possible actions is reduced, making the interface with them simpler, which is very important in the case of applications such as robotic walkers, dedicated to seniors which are not always comfortable with technology.

For instance, Guido is a passive robotic walker propelled by its user, and steering motors appropriety orient its user towards a predefined goal and avoid local obstacles. COOL-Aide also has a steering motor that orients its user to avoid obstacles. Finally, the Passive RT-Walker uses brakes to orient appropriately the person guiding the device. The main drawback of passive robots is that all the weight of the embedded equipment (*e.g.* the steering motor or brakes, the computing ressources, etc.) has to be pushed by the user. In the case of walking assistance devices designed for seniors or of object transportation platforms, it can be an important limitation.

Active robots have actuated propulsion and can move autonomously, which also adds complexity to their interface, the control and the mechanisms (and therefore cost and safety). The CMU Robotic Walker, RT-Walker and SmartWalker are omnidirectional, compared to the others which use differential steering. Some platforms use real rollators as a basis (COOL-Aide, Passive RT-Walker, UTS Assistant), enabling them to move on uneven terrains. Some have suspensions to gain in stability, which can also be provided using tires (*e.g.*, WAR) but with higher energy consumption. Compliance is provided by springs that may be used to ease the control of the device, making it less stiff.

2.2 Interfaces

Table 2.3 presents the ways force-guided robots measure the user's intent for moving the platforms. The most common interface is to use force/torque sensors on the handlebars. They can even be used to detect falls of the user, as done with the SmartWalker, although

Table 2.2 Locomotion modalities of force-guided robots.

Robots name	**Propulsion type**	**Omnidirectional**	**Remarks**
COOL-Aide	Manual	No	None
Guido	Manual	No	None
Passive RT-Walker	Manual	No	Slopes handled
Care-O-Bot II	Actuated	No	None
Gait Assistant	Actuated	No	None
UTS Assistant	Actuated	No	None
Locomaid	Actuated	No	Compliant
MOBIL	Actuated	No	Pneumatic tires
Power-Assisted Walker	Actuated	No	Slopes handled and compliant
WAR	Actuated	No	Uneven floors and slopes handled
AZIMUT-3	Actuated	No	Uneven floors handled
Active RT-Walker	Actuated	Yes	Slopes handled
CMU Robotic Walker	Actuated	Yes	None
SmartWalker	Actuated	Yes	Uneven floors handled

this feature could not be rigorously tested. Figure 2.1 illustrates mechanisms that do not directly use force sensors on handlebars, but they prove to be less effective. Note that all interfaces limit force-sensing to a specific location on the robot.

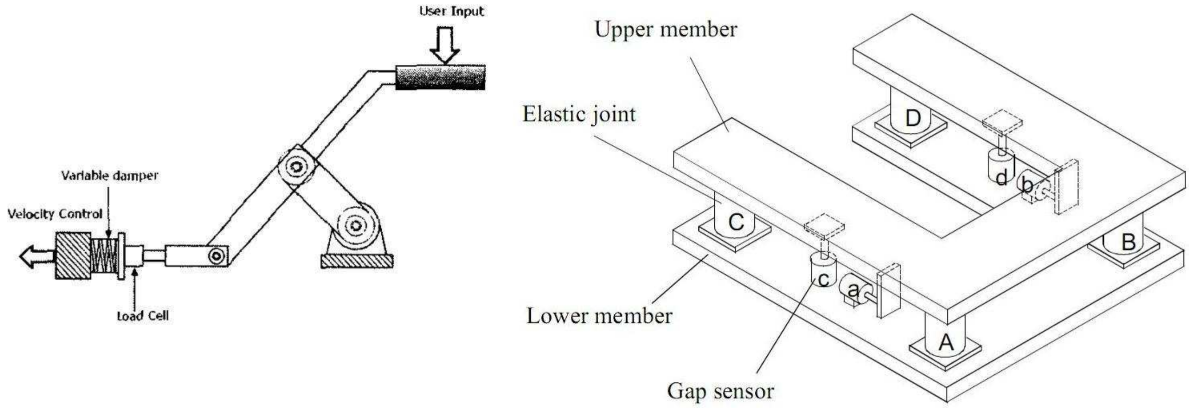

Figure 2.1 Grip lever mechanism [26] (left) and U-shaped supporting arm [8].

Passive platforms that do not need to be turned on and off (*e.g.*, COOL-Aide, Passive RT-Walker) can safely have no other interfaces. These platforms also benefit from the user's preconceived notion of how a classical walker operates. On the other hand, active force-

Table 2.3 Interfaces on force-guided robots.

Robots	**Interfaces type**
Active RT-Walker	Force/torque sensors on the handlebars
Passive RT-Walker	Force/torque sensors on the handlebars
SmartWalker	Force/torque sensors on the handlebars
UTS Assistant	Force/torque sensors on the handlebars
WAR	Force/torque sensors on the handlebars
COOL-Aide	Handlebars with force sensors
Locomaid	Force/torque sensors on the handlebars or a pneumatic bumper
Care-O-Bot II	Handlebars with force sensors, a touch screen which displays destination and state, audio messages for current mode and target
CMU Robotic Walker	Handlebars with two force sensors each, a LCD display which indicates the current desired motion direction
Guido	Handlebar with force sensor, switches to select mode and destination, audio message for navigation events and names of places
MOBIL	Isometric joysticks equipped with two strain gage force sensors to measure traction and compression forces
Gait Assistant	Grip lever mechanism
Power-Assisted Walker	Force sensor embedded on a U-shaped supporting arm
AZIMUT-3	Torque sensors in the orienting wheels

guided robots only equipped with force-torque sensors must develop a special control to handle such situations, which can be as simple as switching off the device when no push is detected for a preset time. Increased functionalities on the platform require more complex interfaces. Platforms that provide paths to goal destination using internal maps often have input and output devices such as speakers or displays. However, such additions increase cognitive and perceptual load for the user, which is not desirable especially for seniors. In addition, seniors often have limited eyesight. As a solution, COOL-Aide (a passive robot) derives user's intent using a map and from force/torque sensory data.

2.3 Force sensing

Table 2.4 presents the approaches used for motion control of force-guided robots. Each robot uses its own specific approach, which can be grouped into three main categories.

Table 2.4 Motion control of force-guided robots.

Robots	**Motion control**
Guido	Unspecified, input is used to steer the device
CMU Robotic Walker	Unspecified custom calculation of velocity based on values measured by force-sensing devices
WAR	Unspecified, custom calculation of velocity based on values measured by force-sensing devices
Care-O-Bot II	Custom calculation of velocity based on values measured by force-sensing devices
Locomaid	Custom calculation of velocity based on values measured by force-sensing devices
Active RT-Walker	Force-torque sensors serve as inputs to a physical model of the device
Passive RT-Walker	Force-torque sensors serve as inputs to a physical model of the device
SmartWalker	Force-torque sensors serve as inputs to a physical model of the device
AZIMUT-3	Torque sensors serve as inputs to a control algorithm that includes a simple kinematic model of the device
Gait Assistant	Force-torque sensors serve as inputs to a simple kinematic model
Power-Assisted Walker	Force-torque sensors serve as inputs to a custom control algorithm
UTS Assistant	Force/torque sensors serve as inputs to a custom velocity calculation approach
MOBIL	Force-torque sensors values set speed setpoints
COOL-Aide	Force-torque sensors serve as inputs to a shared control algorithm which includes a physical model of the device

The first category is to derive the desired velocity based on force-sensing inputs (*e.g.*, Care-O-Bot, Locomaid). The force and torque applied to the platform are directly obtained by summing the output of the force-sensing measurements to have the force in the direction of motion. The orientation of the motion is obtained with the difference of the measurements from the left force sensor with that issued by the right one.

The second category is to use a transfer function that gives to the platform a global apparent mass and damping, as is done with the Active and Passive RT-Walkers and the

SmartWalker. These robots use a damping model of their devices, and apply appropriate control algorithms which take the measured forces as inputs.

An improvement provided by SmartWalker consists in allowing the damping of the device to dynamically change depending on the current velocity of the device. Thresholds empirically defined are brought by this improvement, but tests show that it leads to a lower energy consumption of the device and better user satisfaction.

Finally, the third category consists of merging directly the force applied to a shared-control algorithm, as is done with the COOL-Aide robot, see description in Section 2.7.

2.4 Perception

In robotics, *perception* is the acquisition of knowledge about the robot's environment. This is achieved by extracting information from the robot's sensors, which can be functionally classified into *exteroceptive* or *proprioceptive*. Exteroceptive sensors refer to all sensors that acquire information from the robot's environment (*e.g.*, distance measurements, sound amplitude, etc.). Proprioceptive sensors measure values internal to the robot (*e.g.*, motor velocity, battery voltage, etc.).

As shown in Table 2.5, most mobile force-guided robots developed are equipped with a laser range finder. According to [27] or the CARMEN documentation[1], it is currently very difficult to obtain safe and reliable navigation without such devices. Note that the Power-Assisted Walker and the Active RT-Walker do not allow autonomous navigation, and thus do not have sensors associated.

2.5 Localization

Table 2.6 presents what force-guided robots use for localization, *i.e.*, to determine their position in their operating environment. For the Active and Passive RT-Walker, COOL-Aide and UTS Assistant, there is no localization device, and their sensors are only used to detect obstacles, and perform local collision avoidance. Platforms that are not equipped with laser range finders often have to modify their environment using active or passive beacons, like SmartWalker, Locomaid or MOBIL do, to be able to localize themselves, limiting their use in specific areas. The same is true for robots using preloaded maps, such as Care-O-Bot II and Gait Assistant. Robots using SLAM can operate in unknown and

[1] http://carmen.sourceforge.net/hardware.html

Table 2.5 Exteroceptive sensors of existing force-guided robots.

Robots	**Exteroceptive sensors**
Active RT-Walker	None
Power-Assisted Walker	None
COOL-Aide	An infrared obstacle detection sensor
Locomaid	Ultrasonic range finders and bumpers
MOBIL	An ultrasonic array and a camera
SmartWalker	An ultrasonic array and a camera
AZIMUT-3	A laser range finder
Gait Assistant	A laser range finder
Guido	A laser range finder
Passive RT-Walker	A laser range finder
UTS Assistant	A laser range finder
Care-O-Bot II	A laser range finder and a bumper in front
WAR	A laser range finder and a CCD camera
CMU Robotic Walker	A laser range finder, two circular arrays of ultrasonic transducers, two circular arrays of infrared near-range sensors, and three large touch-sensitive doors

unexplored areas (approximately 1000 m^2). GPS is also an alternative that only works outdoors.

2.6 Navigation

Table 2.7 summarizes the navigation approaches used by force-guided robots. Navigation consists in finding a collision-free path to go from one place to another [4]. This requires a path-planning module to determine a safe path using a map of the environment, and an obstacle avoidance module to handle moving objects and dynamic changes in the world. Robots with no navigation capabilities are for users not visually impaired or who do not have cognitive difficulties. Without path-planning, reactive obstacle avoidance is used to move around perceived obstacles using laser or ultrasonic range finders and bug-type algorithms [4]. Robots such as MOBIL and Gait Assistant are given routes that they follow using a path-tracking algorithm, which can take into account constraints of the user or the platform. However, this approach is not portable in real-world applications where responsiveness of robots is needed. Virtual potential field path-planning algorithms and derivatives are used to derive a path from the current location to a goal location by letting the equivalent of a ball move on a surface representing the environment (with obstacles represented as walls, and the goal location being the lowest point). However, such

Table 2.6 Localization systems of force-guided robots.

Robots	**Localization systems**
Active RT-Walker	None
Passive RT-Walker	None
Power-assisted walker	None
UTS Assistant	None
SmartWalker	Positioning passive optical beacons on the ceiling
Locomaid	Positioning active ultrasonic beacons
MOBIL	Vision-based localization with active optical beacons
AZIMUT-3	A local map built with laser range finder measurements.
Care-O-Bot II	Data fusion of odometry and laser range finder data using Kalman filter on preloaded metric map
Gait Assistant	Data fusion of odometry and laser range finder data using Kalman filter on preloaded metric map
COOL-Aide	A local metric map built with HIMM (Histogramic In Motion Mapping) [2]
Guido	SLAM (Simultaneous Localization And Mapping) : dual version of the FastSLAM algorithm and EKF (Extended Kalman Filter [17]
CMU Robotic Walker	SLAM [38] and fast version of Monte Carlo localization [39]
WAR	GPS (Global Positioning System)

approaches do not consider dynamic obstacles and replanning is often necessary. When a global map of the environment is available, A* and wave-front expansion are algorithms that can select an optimal path between two points. The user must therefore explicitly select a destination, a functionality that requires more complex interfaces.

2.7 Shared-Control

Shared-control refers to the capability of a system to be influenced simultaneously by the user and the robot's control policy. Shared-control involving physical interactions must be safe and must prioritize the user's influences as much as possible. As shown in Table 2.8, shared-control is not present on the majority of force-guided robots: it is either the user who drives the robot, the user controls one parameter (propulsion) and the robot control

Table 2.7 Navigation systems of force-guided robots.

Robots	**Navigation systems**
Active RT-Walker	No path-planning; no obstacle avoidance
Power-assisted walker	No path-planning; no obstacle avoidance
Locomaid	No path-planning; reactive obstacle avoidance
Passive RT-Walker	No path-planning; reactive obstacle avoidance
MOBIL	Predefined path; reactive obstacle avoidance
Gait Assistant	Predefined path and a path tracking algorithm with constraints regarding the platform's kinematic and for smooth user's motion
WAR	Path-planning (no precision given); reactive obstacle avoidance
SmartWalker	Path-planning (no precision given); obstacle avoidance using the distance to obstacles
AZIMUT-3	Local path-planning and obstacle avoidance based on Dynamic Window Approach (DWA)
COOL-Aide	Path-planning and obstacle avoidance using virtual force fields [1], an extended version of the potential field algorithm [35]
UTS Assistant	Path-planning and obstacle avoidance using the VFH (Vector Field Histogram) in the local map [22]
Care-O-Bot II	Wave-front path-planning [19]; obstacle avoidance using elastic bands [16]
Guido	A* in a predefined graph of the environment; obstacle avoidance with replanning on local metric maps

the other (steering), or it is either the robot or the user who exclusively controls the platform at a particular moment (*e.g.*, WAR). Robots implementing shared-control adopt various approaches. For instance, the user of the CMU Robotic Walker defines a desired path on a global map through an interface, and the device slows down as the deviation of this path increases. The interface helps the user to get back on his path. If a change in trajectory from the user occurs, a new destination must be defined. As a solution, Care-O-Bot II compares the desired linear and rotational velocities drawn from the user's input on the force/torque sensors and the planned velocities. If the difference is above an empirically defined threshold, the path is modified automatically by activating behaviors such as wall-following or docking, based on environmental features. However, this approach is complex and requires setting up multiple thresholds, which limits the addition of behaviors. Moreover, if the difference between the desired and planned velocities is too high, Care-O-Bot II switches to a mode where the user has full control of the device. Using a different approach, SmartWalker evaluates the difference between the user's path and a reference path (computed using a global map). Passive RT-Walker interprets information about the environment as forces that are complementary to the forces applied by the user, which are used for the motion control algorithm (Section 2.3). COOL-Aide uses the force applied to propel the device to infer and maintain a goal in the local map. UTS Assistant uses fuzzy logic to merge the outputs of its custom calculation of velocities based on force and the Vector Field Histogram (VFH) algorithm of the autonomous navigation module. Fuzzy logic is particularly appropriate in this case, because it provides a smooth output for inputs that are not always well-defined.

Table 2.8 Shared-control systems of force-guided robots.

Robots	**Shared-control systems**
Locomaid	Unspecified
Active RT-Walker	No shared-control: the user drives the robot
MOBIL	No shared-control: the user drives the robot
Power-assisted walker	No shared-control: the user drives the robot
Gait Assistant	No shared-control: the robot controls the heading, the user controls the velocity
Guido	No shared-control: the user controls propulsion, and the robot controls steering
WAR	No shared-control: the device is either driven or guides its user
CMU Robotic Walker	Three operational modes with different levels of control for the user and the robot
Care-O-Bot II	Path modification based on inferred user's intent
SmartWalker	Shared-control based on an adaptive algorithm
AZIMUT-3	Shared-control mixes user's intent and obstacle avoidance algorithm
Passive RT-Walker	Shared-control based on a motion control algorithm
COOL-Aide	Control is given to the user, unless an obstacle is too close
UTS Assistant	A fuzzy logic component mixes two control outputs

CHAPTER 3

FORCE-GUIDANCE OF AZIMUT-3

The robots presented in Chapter 2 are specifically designed to be force-guided, and require forces to be applied to specific locations on the platform. This can be limiting and can even affect safety of the system when used in the real world. This calls for a system where safety and user satisfaction prevail, using a human-robot interface that is as simple and as natural as possible.

As a solution, we decided to study how the omnidirectional platform AZIMUT-3, shown in Figure 3.1, could be used as a secure, force-guided platform. AZIMUT-3 is a platform developed at IntRoLab[1], the Intelligent, Interactive, Integrated, Interdisciplinary Robotic Laboratory of the Université de Sherbrooke. It uses four propulsion and steerable wheels called *AZIMUT wheels*. Each of these wheels is made of a backdrivable DC brushless motor for propulsion and a Differential Elastic Actuator (DEA) for steering. The DEA, also developed at IntRoLab, can be controlled in position, velocity and torque, making it compliant. It provides controlled mechanical elasticity by reducing internal mechanical efforts through the presence of a spring element. The DEA is conceptually similar to Series Elastic Actuator (SEA) [30, 42], but uses a differential coupling instead of a serial coupling between a high impedance mechanical velocity source and a low impedance mechanical spring. This results in a more compact and simpler solution, with similar performances. Using DEA for wheel steering on AZIMUT-3 gives the platform a unique elastic behavior, or a kind of horizontal suspension, and when pushed, the forces and torques sensed through DEA can be used to derive an intended direction, and therefore assist user's motion. Sensing forces and torques directly from the steered axes makes it possible to apply forces on all the robot's structure (and not just on instrumented handlebars). The challenge is to derive the intended force and torque applied on the platform from the forces and torques measured on the four steering axes. AZIMUT-3 can be equipped with laser range finders, making it possible to implement navigation, obstacle avoidance and, consequently, shared-control capabilities.

To conduct this study, our methodology consisted in first validating our approach in simulation before initiating trials on the real robot. This allowed us to safely and rapidly

[1] http://introlab.gel.usherbrooke.ca/mediawiki-introlab/index.php

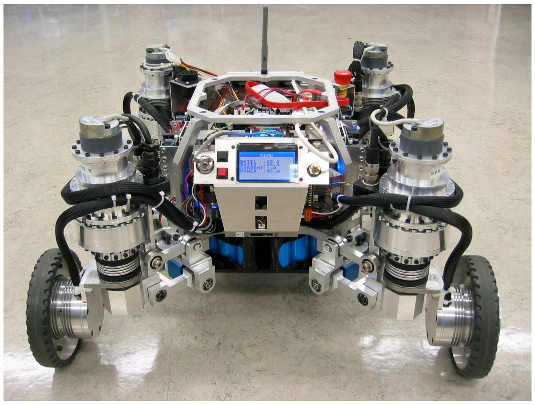

Figure 3.1 AZIMUT-3.

experiment ideas and concepts. Webots[2] is used because of its precise physical simulations (*i.e.*, computing forces and torques) using ODE (Open Dynamics Engine[3]). Its architecture also makes it possible to use the same C++ code between the simulator and the real platform, and also to be interfaced with existing libraries such as ROS (Robotic Operating System[4]) that is used on AZIMUT-3.

[2] http://www.cyberbotics.com/

[3] http://www.ode.org/

[4] http://www.ros.org/wiki/

CHAPTER 4

SECURED FORCE-GUIDANCE OF THE OMNIDIRECTIONAL NON-HOLONOMIC PLATFORM AZIMUT-3

4.1 Foreword

Authors and affiliations:

J.Frémy received his Master's of Science in electronics, automatism and computer science at the Ecole Supérieure d'Electronique de l'Ouest (ESEO) of Angers, France in 2009. Since 2007, he is a candidate for a Master's degree in Electrical Engineering at the Université de Sherbrooke, Québec, Canada.

F.Michaud (M'90) received his bachelor's degree ('92), Master's degree ('93) and Ph.D. degree ('96) in Electrical Engineering from the Université de Sherbrooke, Québec, Canada. After completing postdoctoral work at Brandeis University, Waltham MA ('97), he became a faculty member in the Department of Electrical Engineering and Computer Engineering of the Université de Sherbrooke, and founded IntRoLab (formerly LABORIUS), a research laboratory working on designing intelligent autonomous systems that can assist humans in living environments. Prof. Michaud is the Canada Research Chairholder in Mobile Robots and Autonomous Intelligent Systems, and the Director of the Interdisciplinary Institute for Technological Innovation. He is a member of IEEE, AAAI and OIQ (Ordre des ingénieurs du Québec). In 2003, he received the Young Engineer Achievement Award from the Canadian Council of Professional Engineers.

M. Lauria received his M.Sc. in Micro-engineering in 1997 and his Doctoral degree in 2003 from the Swiss Federal Institute of Technology (EPFL). From 2003 to 2009, he was Professor at the Department of Electrical and Computer Engineering of the Université de Sherbrooke, Québec, Canada. Since 2009, he is Professor at the University of Applied Sciences Western Switzerland (HES-SO).

Submission date: November 16, 2010.

Journal: IEEE Transactions on Robotics.

Contribution: In the article we describe our approach to the research problem introduced in Chapter 3. Therefore we present an algorithm to derive the user's physical intent from the torques measurements on the wheel axis of a robot. We also present a shared control method that determines a safe but as close as possible from user's intent using a laser range finder. Tests are also presented to validate these contributions. Further information on the calculation introduced in Section 4.5.1 are presented in Appendix A, and more details on the implementation of the architecture evoked in Section 4.5 in Appendix B.

French title and abstract: Contrôle en force de la plateforme non holonome AZIMUT-3.

Pouvoir guider et être guidé physiquement comme une personne est une capacité qui serait intéressante pour les robots, pour les aides à la mobilité par exemple, ou encore permettre une interaction humain-robot naturelle. Une solution est de placer des capteurs en force à des endroits précis sur le robot pour détecter l'intention de l'utilisateur, mais cela limite les emplacements où ces interactions peuvent avoir lieu. Une alternative consiste à utiliser les données de couple de roues orientables sur une plateforme omnidirectionnelle non-holonome. Cela permet de percevoir les forces appliquées sur la plateforme directement via son mécanisme de locomotion. Cet article présente une approche pour guider en force AZIMUT-3, un robot équipé de moteurs différentiels élastiques et contrôlés en couple pour orienter ses roues tout en restant en mouvement. Les résultats de tests en situation réelle montrent que notre approche permet de déplacer AZIMUT-3 en réponse à des commandes physiques données par un être humain poussant le robot tout en faisant éviter à ce dernier les obstacles et collisions.

ABSTRACT

Physically guiding someone or being guided is an interaction capability that would be beneficial for robots, for instance for mobility assitances, or allowing a natural human-robot interaction. One solution is to place force sensors at specific locations on the robot to detect the user's intent, but this limits where physical interaction can occur. An alternative consists in using torque data from steerable wheels of an omnidirectional non-holonomic platform, making it possible to perceive forces applied on the platform through its locomotion mechanism. This paper presents an approach to force-guide AZIMUT-3, a mobile robot equipped with backdrivable and torque controlled differential elastic actuators for active wheel steering. Real world results demonstrate that our approach allows AZIMUT-3 to move in response to physical commands given by a human pushing it while avoiding obstacles and collisions.

4.2 INTRODUCTION

To make robots move from industrial to natural settings, they must be able to interact safely and naturally in direct physical contact with people. However, most mobile robots are still actuated with motors that are not backdrivable. Thus, when a contact occurs between the robot and an object or a human being, both the motors and the encountered entity must be able to sustain the shock. Yet, humans use direct physical interactions to influence their motion. For instance, guiding someone by holding his/her hand or the shoulders is very common. Such natural interface would be beneficial for mobile robots too, instead of relying on remote controllers (*e.g.*, joysticks, gamepads) or having to physically carry the robot. In such a scenario, the robot's motors should be put to use for moving in the direction given by someone physically guiding the robot. Such a platform must be able to safely support physical contacts, and respond appropriately. In addition, the robot could still use its sensors to guarrantee motion safety. Such capabilities are inherently applicable to mobile devices such as motorized carts, electric wheelchairs, etc.

This paper presents an omnidirectional mobile platform (AZIMUT-3), that can detect forces on the horizontal plane and move in the intended direction [10, 11]. To do so, the platform resorts to orientable wheels motorized using differential elastic actuators (DEA) [20], which provide compliance, safety and torque control capabilities. This design provides a natural physical interface without requiring the use of costly sensors such as six Degrees

Of Freedom (DOF) force/torque sensors. In addition, lidar data are used by the control algorithm to safely guide AZIMUT-3 by physical interaction.

The paper is organized as follows. Section 4.3 presents an overview of systems guided through physical interactions, such as object-transportation and walking assistant devices. Section 4.4 introduces our platform and its characteristics. Section 4.5 describes its control, allowing it to sense the forces through the DEAs and to take into account lidar measures to generate commands in agreement with user's intent and safeness. Finally, section 4.6 presents the results obtained in real-world trials, demonstrating the feasibility of the concept.

4.3 Force-Guided Mobile Platforms

To date, force-guided mobile platforms are either motorized carts or robotic walkers. Table 4.1 summarizes these platforms according to the following characteristics:

- Passive (P) / Active (A) platforms: a passive platform can steer its joints but requires a human to propel it, limiting its usage and the equipment it can carry. An active platform has propulsion which makes it possible to assist user's motion.
- Omnidirectionality: an omnidirectional platform can move in all directions without changing its orientation, providing better maneuverability.
- Obstacle avoidance: the platform is equipped with range sensors and can react to obstacles.
- Shared-control: the platform combines user's intent with navigation data to control motion. We consider control to be shared when influences derived from the user and from other sources (*e.g.*, obstacle detection) contribute simultaneously to derive the actuation commands. A special case, identified as *Sep*, happens when user inputs are used to control *separated* degrees of freedom (*e.g.*, user inputs control the robot's propulsion and proximity sensory data influence steering).

For the active platforms listed in Table 4.1, force and torque sensors are integrated in handle bars placed on the platforms, limiting the application of forces to specific locations on the robot. This requires sophisticated force/torque sensors and control systems. Training is also required to allow users to learn how to operate the platform, limiting the simplicity of direct physical interaction with the platform. In addition, only a few of these platforms have shared-control capabilities. The Passive RT-Walker and Active RT-Walker use a damping model of their platforms and apply a control algorithm that takes the desired forces as inputs. However, large forces and torques have to be applied to reach areas

Table 4.1 Force-guided mobile platforms.

Platform	Passive (P) Active (A)	Omnidirec- tionality	Obstacle avoidance	Shared- control
COOL-Aide [40]	P	No	Yes	Sep
Guido [18]	P	No	Yes	Sep
Passive RT-Walker [14]	P	No	Yes	Yes
Power-Assisted Walker [8]	A	No	No	No
Locomaid [24]	A	No	Yes	N/A
RoTa [33]	A	No	Yes	No
MOBIL [32]	A	No	Yes	No
Care-O-Bot II [12]	A	No	Yes	Yes
Gait Assistant [26]	A	No	Yes	No
WAR [34]	A	No	Yes	No
CMU Robotic Walker [25]	A	Yes	Yes	Yes
Active RT-Walker[6]	A	Yes	Yes	Yes
SmartWalker [37]	A	Yes	Yes	Yes
UTS Assistant [22]	A	No	Yes	Yes

close to obstacles and stairs because virtual forces computed in these cases are high. For Care-O-Bot II, shared-control consists of activating a behavior such as wall-following or docking when the difference between user intent and obstacle avoidance control is above a threshold. However, this requires complex logic and the definition of behavior activation thresholds, and complicates the addition of new behaviors. The CMU Robotic Walker uses a predefined path to a target to influence user's motion: if the user goes away from the predefined path, the platform slows down to eventually come to a stop, waiting for the user to push it towards the predefined path again. Similarly, SmartWalker conducts an online evaluation of the user's performance, partly based on the difference between the predefined path and the actual path, and gives more or less control to the user based on this evaluation. These last two approaches complexify motion control because the user has to explicitly communicate to the robot the predefined path toward the intended target. The UTS Assistant merges the intended and desired velocities using a fuzzy logic controller, which however requires to optimize multiple membership functions for each user.

With AZIMUT-3, our objective is to demonstrate force-guiding capabilities of an omnidirectional platform not limited to sensing forces at specific locations (*e.g.*, handle bars or joysticks), using a novel and safe and shared-control approach.

4.4 AZIMUT-3, an Omnidirectional Platform with Force-Controlled Steerable Wheels

AZIMUT-3, shown in Fig. 4.1, is an omnidirectional, non-holonomic four-wheel steerable platform. It comes equipped with a Mini-ITX, 2.0 GHz Core 2 duo processor running Linux with real-time patches (RT-PREEMPT). Its hardware architecture is made of distributed modules for sensing and low-level control, communicating with each other through a 1 Mbps CAN bus [23]. The platform has a 34 kg payload and can reach a maximum velocity of 1.47 m/s. Nickel-metal hybrid batteries provide power to the platform for about half an hour autonomy at maximum speed.

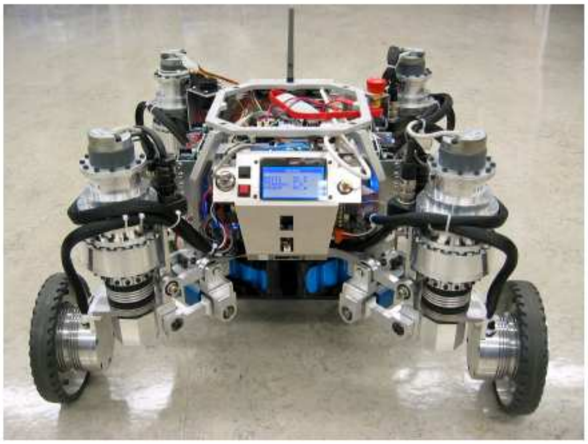

Figure 4.1 AZIMUT-3 platform.

Omnidirectionality of AZIMUT-3 is provided by steerable and drivable wheels with a lateral offset from its attachment point (referred to as *AZIMUT Wheels* [21]). Compared to other wheels used on omnidirectional platforms, they are lighter and mechanically simpler than Swedish wheels and provide a built-in horizontal suspension system absent when active caster wheels are used. They also permit to lower the chassis' height. A passive vertical suspension made of four Rosta springs is used to connect the steerable wheels to AZIMUT-3's chassis, allowing the wheels to keep contact with the ground on uneven surfaces.

AZIMUT-3's wheels are each made of a propulsion actuator and a steer actuator. The propulsion actuator consists of a DC brushless motor (K064050-3Y from Bayside) and a wheel encoder (E4-300 from US Digital, resolution 0.3 deg). Steering is done using DEAs. A DEA is conceptually similar to a Series Elastic Actuator (SEA [42][30]), but uses a differential coupling instead of a serial coupling between a high impedance mechanical speed source and a low impedance mechanical spring. This results in a more compact and simpler solution, with similar performances. The DEA used for steering is made of a DC brushless motor (K064050-7Y-2 from Bayside), a wheel encoder (RM44 from Renishaw, resolution 0.2°) and a reaction torque sensor (TRT-500 from Transducer Techniques).

It is possible to control the DEA's mechanical elasticity and viscosity in accordance with the admittance control scheme [15] expressed by (4.1):

$$\frac{X(s)}{F(s)} = \frac{1}{Ds + K} \tag{4.1}$$

where F is the force sensed at the output of the DEA, D and K are the desired apparent damping and stiffness, and X is the measured DEA's steer angle which determines the orientation of the wheel. This makes the DEA act as an active elastic element that can inherently absorb shocks, perceive the forces from the environment on the robot and control the forces applied back to the environment.

Each wheel can be steered over a 180° range, as shown by the dotted lines in Fig. 4.2, to avoid interferences with the robot's chassis. To make AZIMUT-3 move, all wheels must be precisely coordinated [7]: they must all be oriented in the same direction, or have all their axis converge toward one point called the Instantaneous Center of Rotation (ICR) of the platform. This results in three possible modes of locomotion [21], also illustrated in Fig. 4.2 :

- Mode 1: ICR is located in the triangular regions (the right one when the robot is turning right, or the left one when the robot is turning left) on the sides of the platform but not next to it. This mode is used when the robot has to move forward or do slight turns (i.e., high radius of curvature). When all wheels are parallel, ICR tends toward infinity.
- Mode 2: similar to Mode 1 but for moving sideways.
- Mode 3: ICR is located near the platform's chassis, which allows the robot to make tight turns (i.e., turns with low radius of curvature) and rotate on itself.

Due to physical limitations of the steering axis, changing modes require the platform to come to a stop. For instance, Fig. 4.3 illustrates the situation when the platform decreases its turning radius, resulting in a change of its locomotion mode from Mode 1 to Mode 3: as the turning radius decreases, the wheels must change their orientation. Elasticity in wheel steering actuation provide safe and robust coordination of the wheels.

Fig. 4.2 also illustrates the constraints we imposed on the ICR in each mode to limit the complexity of our approach. Because DEAs can only detect torques from forces creating a moment on the wheel's axis, for a given translational direction (in x or y), lateral forces

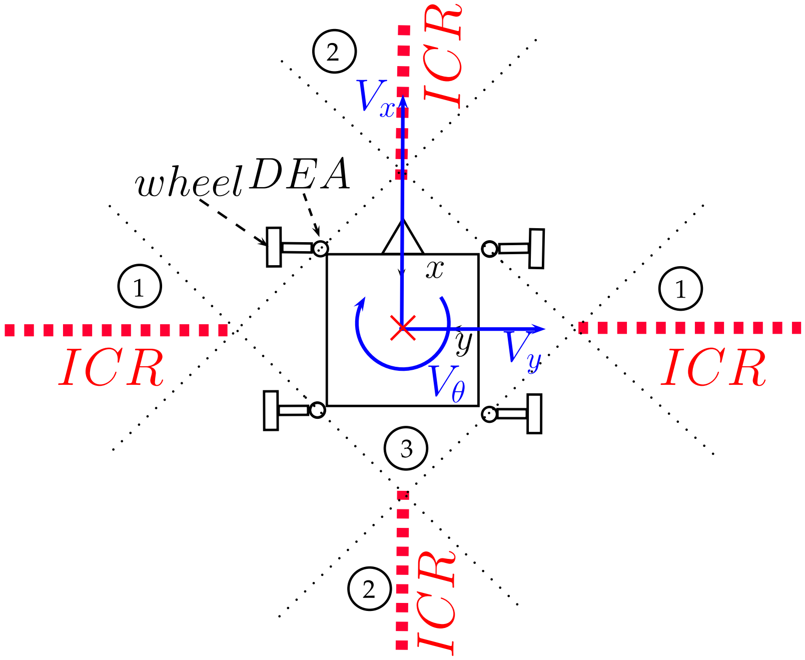

Figure 4.2 Top view representation of AZIMUT-3. Thin dashed lines represent Modes' borders. The ICR can only be in one of the numbered regions. Bold dashed lines represent the restricted space reachable by the ICR in our approach.

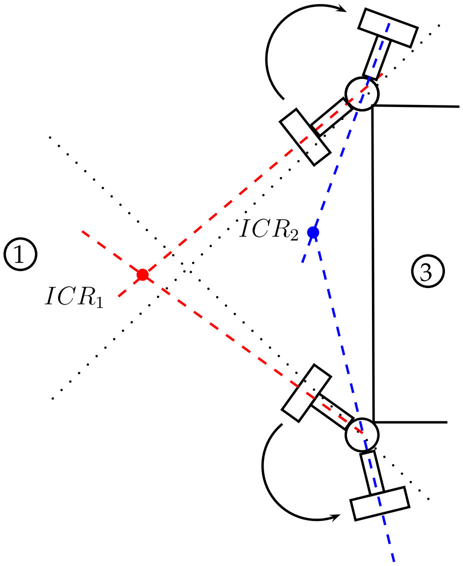

Figure 4.3 ICR defined in Mode 1 and Mode 3. Close ICRs in the plane of motion can create discontinuities in the orientation of the wheels.

cannot be detected. This occurs for Mode 1 and Mode 2 in ICR moving on the dashed lines, making AZIMUT-3 move similarly to shopping carts or rollators. For Mode 3, ICR is restrained to be at the center of the platform (identified by $\times$), allowing the platform to rotate in place.

4.5 Force-Guidance of AZIMUT-3

Safe force-guidance of AZIMUT-3 requires to derive push intent from the forces sensed through the DEAs to control wheel steering and propulsion of the platform, and to ensure the safety of the platform by avoiding obstacles and collisions. In this work, we assume that AZIMUT-3 operates on flat surfaces (*i.e.*, gravity is not considered), that the DEAs' admittance control scheme is fixed (*i.e.*, D and K are kept constant), and forces that are collinear with the wheels' propulsion axis are not considered.

The control architecture is illustrated in Fig. 4.4, and detailed explanations are provided in the following subsections. To briefly explain the process, DEAs provide raw torque readings $T'_{0\to i}$ (the torque T from the chassis 0 to the DEA) and angular position β of wheel i, from which the force F'_{APP} and the torque T'_{APP} applied on the chassis are estimated. To control AZIMUT-3's motion, the Mode (as described in Section 4.4) and the Twist must be determined. A Twist is a vector containing the translational velocity V_x, the lateral velocity V_y and the rotational velocity V_θ of the robot in the plane. The Motion Assistance module determines the maximum Twist commands according to the force and torque applied by the user to the platform. The Push Intent module derives a goal g from the force and torque intent of the user, to let the Local Path Planner module determine a Twist that also takes into account obstacle avoidance using a local occupancy map. Finally, the Motion Control module, described in [3], converts Twist and mode variables into applicable motor commands for AZIMUT-3.

4.5.1 Applied Force and Torque Identification

The first step in our approach is to derive the applied force and torque on the platform from the forces and torques perceived on each wheel.

First, torques $T_{0\to i}$ on each wheel is evaluated. This requires to filter the sensor noise of each wheel using a second order Chebychev low pass filter $H(s)$ with a normalized cutoff frequency of 0.05 empirically defined, as expressed by Equation (4.2).

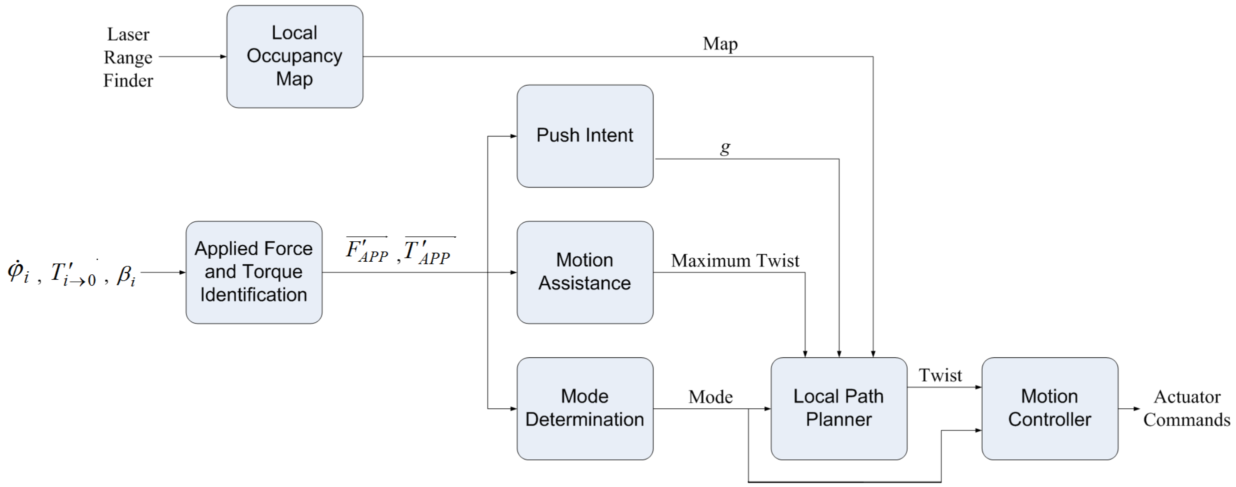

Figure 4.4 Force-guidance control architecture.

$$T_{0\rightarrow i-chebychev}(s) = T'_{0\rightarrow i}(s).H(s) \tag{4.2}$$

Effects of the dynamics on the measured torque are approximated by a linear function proportional to wheel acceleration $\ddot{\phi}_i$, as expressed in Equation (4.3). These accelerations are obtained from the time derivative of the measured velocity of rotation of each wheel $\dot{\phi}_i$ and filtered with a second order low pass filter. C_{acc} has been determined empirically by measuring DEA's torque with AZIMUT-3 moving at various velocities. For each wheel, it is similar to the wheel's inertial moment, but considers also the coupling between the DEA and the chassis. To ignore residual torques that can arise from DEA's harmonic drive, a dead zone is set using an empirical threshold T_{dz} to derive $T_{0\rightarrow i}$.

$$T_{0\rightarrow i-inertial} = T_{0\rightarrow i-chebychev} - C_{acc}.\ddot{\phi}_i \tag{4.3}$$

$$T_{0\rightarrow i} = \begin{cases} T_{0\rightarrow i-inertial} & \text{if } |T_{0\rightarrow i-inertial}| > T_{dz}; \\ 0 & \text{if } |T_{0\rightarrow i-inertial}| < T_{dz}. \end{cases} \tag{4.4}$$

Equations (4.2), (4.3) and (4.4) therefore express the filtering process used to derive torque $T_{0\rightarrow i}$.

Second, Newton's second law is used to derive a model of AZIMUT-3's chassis. As expressed by (4.5) and (4.6), the forces and the sum of the torques applied to the chassis are null. Fig. 4.5 illustrates the elements used in this model. P is the center of the square

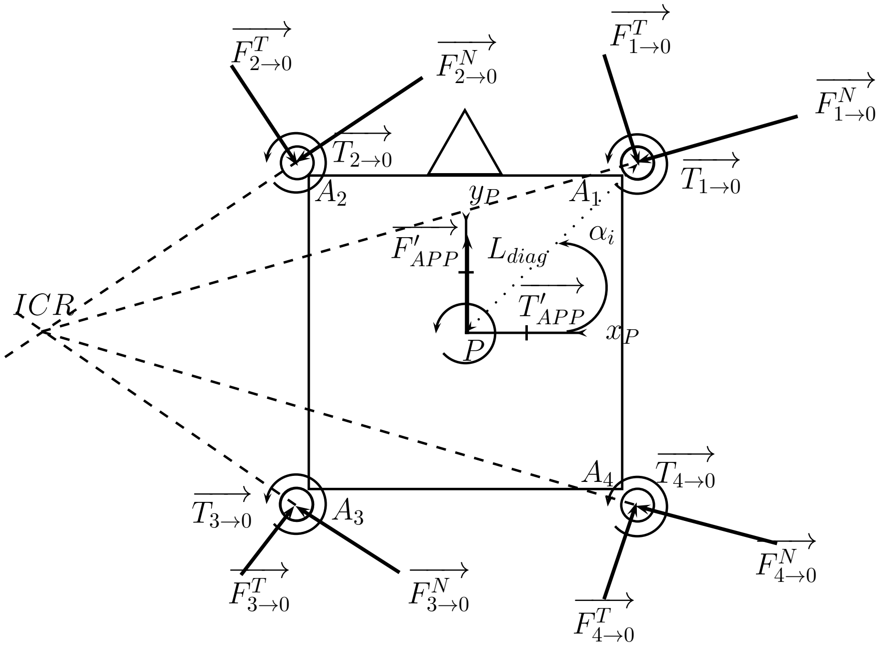
$\overrightarrow{F^T_{2\to 0}}$
$\overrightarrow{F^N_{2\to 0}}$
$\overrightarrow{F^T_{1\to 0}}$
$\overrightarrow{F^N_{1\to 0}}$
$\overrightarrow{T_{2\to 0}}$
$\overrightarrow{T_{1\to 0}}$
A_2
y_P
A_1
$\overrightarrow{F'_{APP}}$
L_{diag}
α_i
ICR
$\overrightarrow{T'_{APP}}$
x_P
P
A_4
$\overrightarrow{T_{4\to 0}}$
$\overrightarrow{T_{3\to 0}}$
A_3
$\overrightarrow{F^N_{4\to 0}}$
$\overrightarrow{F^T_{3\to 0}}$
$\overrightarrow{F^N_{3\to 0}}$
$\overrightarrow{F^T_{4\to 0}}$

Figure 4.5 Force and torque model of AZIMUT-3's chassis. A_i refers to the center of the DEAs, represented by circles.

formed by the chassis, A_i is the center of the DEA belonging to the ith wheel. L_{diag} is the distance between P and A_i, which is the same for each i.

$$\overrightarrow{0} = \overrightarrow{F'_{APP}} + \sum_{i=1}^{4} \overrightarrow{F_{i\to 0}} \tag{4.5}$$

$$\overrightarrow{0} = \overrightarrow{T'_{APP}} + \sum_{i=1}^{4} \overrightarrow{T_{i\to 0}} + \sum_{i=1}^{4} \overrightarrow{PA_i} \times \overrightarrow{F_{i\to 0}} \tag{4.6}$$

where

$$\overrightarrow{F_{i\to 0}} = \overrightarrow{F^T_{i\to 0}} + \overrightarrow{F^N_{i\to 0}} \quad i = 0...4 \tag{4.7}$$

Newton's second law can also be applied to a single wheel, as expressed by (4.8) and (4.9) in the $\mathcal{F}_P$ (axis x_P and y_P) reference frame. Fig. 4.6 illustrates the elements of this model. A_i is the position of wheel i and B_i is the projection of the wheel's center of gravity in the plane which includes A and P.

L_{arm} is the length between A_i and B_i, which is the same for each wheel. GND stands for ground and is used to point out forces coming from the ground.

$$\overrightarrow{0} = \overrightarrow{F_{0\to i}} + \overrightarrow{F_{GND\to i}} \tag{4.8}$$

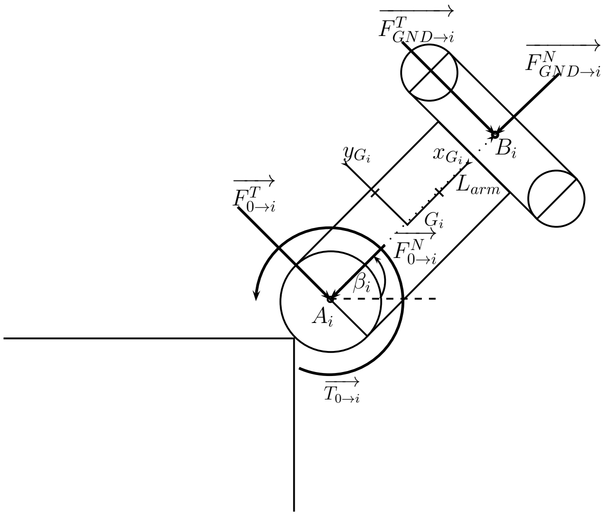
$\overrightarrow{F^T_{GND\to i}}$
$\overrightarrow{F^N_{GND\to i}}$
B_i
y_{G_i}
x_{G_i}
L_{arm}
$\overrightarrow{F^T_{0\to i}}$
G_i
$\overrightarrow{F^N_{0\to i}}$
β_i
A_i
$\overrightarrow{T_{0\to i}}$

Figure 4.6 Force and torque model of wheel i.

$$\overrightarrow{0} = \overrightarrow{T_{0\to i}} + \overrightarrow{G_iA_i} \times \overrightarrow{F_{0\to i}} + \overrightarrow{G_iB_i} \times \overrightarrow{F_{GND\to i}} \tag{4.9}$$

where

$$\overrightarrow{F_{0\to i}} = \overrightarrow{F^T_{0\to i}} + \overrightarrow{F^N_{0\to i}} \tag{4.10}$$

$$\overrightarrow{F_{GND\to i}} = \overrightarrow{F^T_{GND\to i}} + \overrightarrow{F^N_{GND\to i}} \tag{4.11}$$

Combining (4.5) and (4.8) for the forces and (4.6) and (4.9) for the torques, vector $\overrightarrow{F_{0\to i}}$ in the frame $\mathcal{F}_{G_i}$ defined by G_i (axis x_{G_i} and y_{G_i}) is expressed by (4.12). Equation (4.12) is valid in both frames $\mathcal{F}_P$ and $\mathcal{F}_{G_i}$ because their z-axes are parallel.

$$^{\mathcal{F}_{G_i}}\overrightarrow{F_{0\to i}} = \begin{pmatrix} 0 \\ -\frac{T_{0\to i}}{L_{arm}} \\ 0 \end{pmatrix} \tag{4.12}$$

with

$$\overrightarrow{T_{0\to i}} = \begin{pmatrix} 0 \\ 0 \\ T_{0\to i} \end{pmatrix} \tag{4.13}$$

The known angular position β_i of the wheels set the rotation between $\mathcal{F}_{G_i}$ and $\mathcal{F}_P$, and therefore $^{\mathcal{F}_P}\overrightarrow{F_{0\to i}}$. Using Newton's third law, $^{\mathcal{F}_P}\overrightarrow{F_{i\to 0}}$ and $\overrightarrow{T_{i\to 0}}$ can be defined as:

$$^{\mathcal{F}_P}\overrightarrow{F_{i\to 0}} = -^{\mathcal{F}_P}\overrightarrow{F_{0\to i}} \tag{4.14}$$

$$ {}^{\mathcal{F}_P}\overrightarrow{T_{i\rightarrow 0}} = -{}^{\mathcal{F}_P}\overrightarrow{T_{0\rightarrow i}} \tag{4.15}$$

Using (4.5) and (4.6), $\overrightarrow{F'_{APP}}$ and $\overrightarrow{T'_{APP}}$ can be expressed by (4.16) and (4.17).

$$\overrightarrow{F'_{APP}} = \begin{pmatrix} F'_{APP_x} \\ F'_{APP_y} \\ 0 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{4} -\frac{T_{0\rightarrow i_z}}{L_{arm}}.sin(\beta_i) \\ \sum_{i=1}^{4} -\frac{T_{0\rightarrow i_z}}{L_{arm}}.cos(\beta_i) \\ 0 \end{pmatrix} \tag{4.16}$$

$$\overrightarrow{T'_{APP}} = \sum_{i=1}^{4} \overrightarrow{T_{0\rightarrow i}} - \sum_{i=1}^{4} \begin{pmatrix} 0 \\ 0 \\ -L_{diag}.\frac{T_{0\rightarrow i_z}}{D}.cos(\alpha_i + \beta_i) \end{pmatrix} \tag{4.17}$$

Note that when all wheels are parallel and moving forward, $sin(\beta_i)$ equals 0. Thus the lateral force F'_{APP_x} cannot be detected. This explains why lateral forces are not considered with this model.

4.5.2 Mode Evaluation

Fig. 4.7 illustrates the Finite State Machine (FSM) used to control mode changes according to the constraints presented in Section 4.4. At initialization, the platform is in Mode 3 (with V_x and V_y null), from which Mode 1 and Mode 2 are reachable. When a force is applied and exceeds an empirical threshold C_{t_f}, AZIMUT-3 goes into Mode 1 or Mode 2 depending on force direction (x or y). The robot remains in the selected mode until an inactivity counter c_{in}, which increases when no forces or torques are perceived, reaches a pre-determined period C_{in}, making the robot go back into Mode 3.

4.5.3 Motion Assistance

The Motion Assistance module implements a controller that makes the platform operate like an object with a translational damping D_{TTF} and mass M_{TTF}, and an inertia J_{RTF} and rotational damping D_{RTF}. This is similar to the approach used by SmartWalker [37] and Walking Helper [6]. The mechanical characteristics given to the platform are bounded by stability limits. The controller is expressed by (4.18) and (4.19), where V is the translational velocity and V_θ is the rotational velocity.

$$\frac{V(s)}{F'_{APP_y}(s)} = \frac{1}{M_{TTF}s + D_{TTF}} \tag{4.18}$$

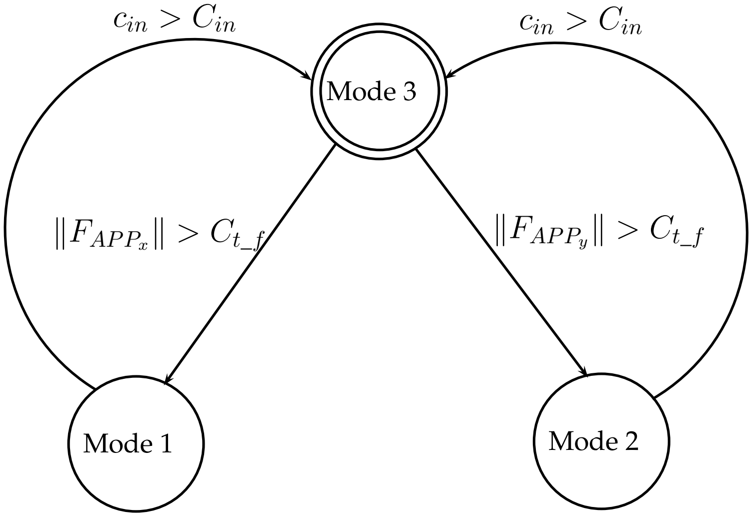

Figure 4.7 FSM for Mode Evaluation.

$$\frac{V_\theta(s)}{T'_{APP}(s)} = \frac{1}{J_{RTF}s + D_{RTF}} \tag{4.19}$$

Thus, depending on the modes, Twist commands are determined as follows:

- In Mode 1, V_x equals V in (4.18), V_y equals 0.
- In Mode 2, V_x equals 0, V_y equals V in (4.18).
- In Mode 3, V_x equals 0, V_y equals 0.

4.5.4 Push Intent

The Push Intent module derives a goal position based on F'_{APP} and T'_{APP} and AZIMUT-3's constrained motion (as explained in Section 4.4). As illustrated in Fig. 4.8, the force applied in the forward direction F'_{APP_y} and the torque applied T'_{APP} determine the polar coordinates (ρ_g, θ_g) of the goal point using (4.20) and (4.21):

$$\rho_g = C_{\rho_g}.F'_{APP_Y} \tag{4.20}$$

$$\gamma_g = C_{\gamma_g}.T'_{APP} \tag{4.21}$$

The trajectory to the determined goal describes an arc of circle going through the robot's center. AZIMUT-3 can follow this trajectory since the ICR of the trajectory lies on the space reachable by the ICR illustrated by Fig. 4.2. The path points are placed at regular intervals on this arc of circle. Coefficients C_{ρ_g} and C_{γ_g} are determined empirically (see Section 4.6 to have their actual values) to establish a good compromise between safety (*i.e.*, velocities suitable versus the platform's distance from obstacles) and user's control of the device (*i.e.*, the platform goes where the user actually wants to go).

4.5.5 Local Occupancy Map and Local Plan Planner

These two modules implement our shared-control mechanism. Fig. 4.9 illustrates their role with two simple cases. When the platform is directed toward a wall, no correction is required. However, when the platform is directed to go through a door, the objective is to provide a small correction to avoid a collision with the edge of the door. These modalities are only operational in Mode 1, because the laser range finder only faces forward.

The Local Occupancy Map module filters laser range finder data using a median filter to derive a local 2D occupancy map, as the one shown in Fig. 4.10. Black cells represent obstacles, white cells are unoccupied, grey cells are unknown, and the robot is the black cell located at the center. Resolution is set based on a trade-off between accuracy, safety and processing power.

The Local Path Planner module implements an adaptation of the Dynamic Window Approach (DWA) [9]. It simulates several possible paths and evaluates them using the local 2D occupancy map. Normally, paths with DWA are derived using a discretization of the maximal velocities. In our particular case, paths are only composed of a translational and a rotational components because of the constraints on Mode 1. In addition, the translational component is always positive because this module is only active when the platform is going in the forward direction. The maximum velocities are set using the Twist command determined by the Motion Assistance module.

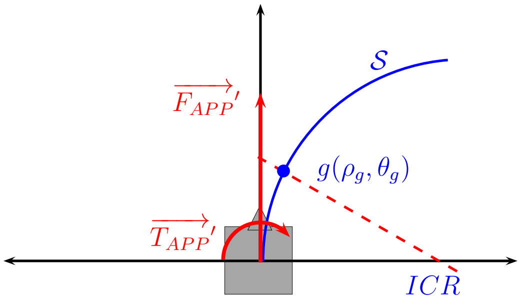

Figure 4.8 Push intent derived from F'_{APP} and T'_{APP}.

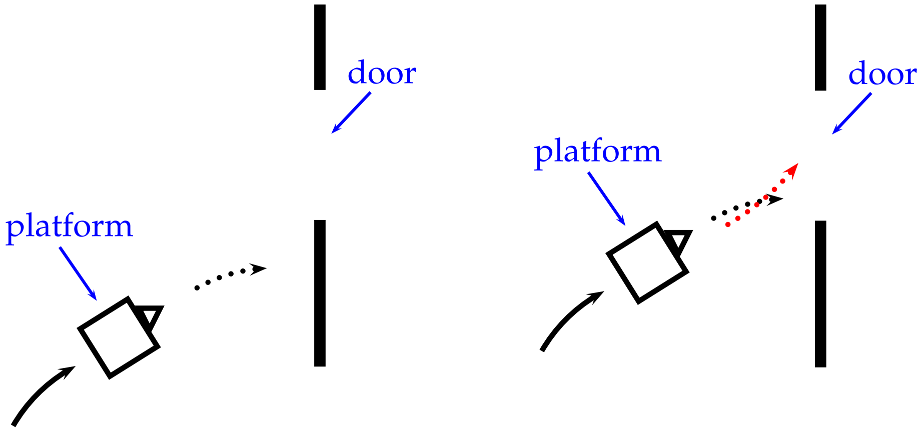

Figure 4.9 Illustrations of cases when (left) the platform is directed to be placed in front of a wall, and (right) the platform is directed to go through a door.

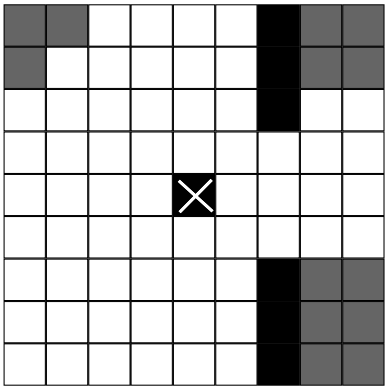

Figure 4.10 Local 2D occupancy map.

Paths with obstacles are eliminated, and paths free of obstacles are evaluated using a cost function C, given by (4.22), with d_goal being the distance between the current position and the goal g, d_path the distance between the current position and the path points. C_{d_goal} emphasizes distance reduction between the final point of the candidate trajectories and the actual goal. C_{d_path} is used to minimize the distance between the path points of the candidate trajectories and the planned path. The last element of the cost function C adds a component proportional to the inverse of the translational velocity, to favor faster trajectories. The velocity V_x is also a number which can be positive or null. To avoid the singularity if V_x is nil, we always add 1 in the cost calculation as expressed by Equation (4.22).

$$C = C_{d\_goal}.d\_goal + C_{d\_path}.d\_path + C_{speed}.\frac{1}{V_x + 1} \tag{4.22}$$

The trajectory with the lowest cost is the one that brings the platform as close to the goal as possible, the fastest and according to the planned path. The cost function provides a good trade-off between user's intent, in terms of location and velocity, and safe navigation by rejecting paths leading to collisions. A Twist command is finally derived from the current position and the first path point of the trajectory with the lowest cost.

4.6 Results

We implemented our force-guidance approach on AZIMUT-3 using the Robotic Operating System (ROS) framework [28]. Table 4.2 presents the parameters introduced in Sections 4.4 and 4.5 used in our trials.

Figure 4.11 illustrates our experimental testbed using AZIMUT-3. To provide comparative data for evaluating the validity of our approach, we installed a 6 DOF force/torque sensor (TW-MINI45 from ATI Industrial Automation, 6500$US) on the handle bar's pole. This sensor provides the force and torque applied on it in all directions and around all axes. It is placed close to the point P introduced in Section 4.5.1. Thus, the transversal force along the forward direction F_{APP} and the torque around platform's pole T_{APP} can be measured, making it possible to compare with F'_{APP} and T'_{APP}, derived using torque data from each wheel.

Two sets of trials were conducted: 1) participants made the platform follow a desired path drawn on the ground, with no obstacles and as fast as possible; 2) participants were blindfolded and had to push the platform through an obstacle course, to evaluate the

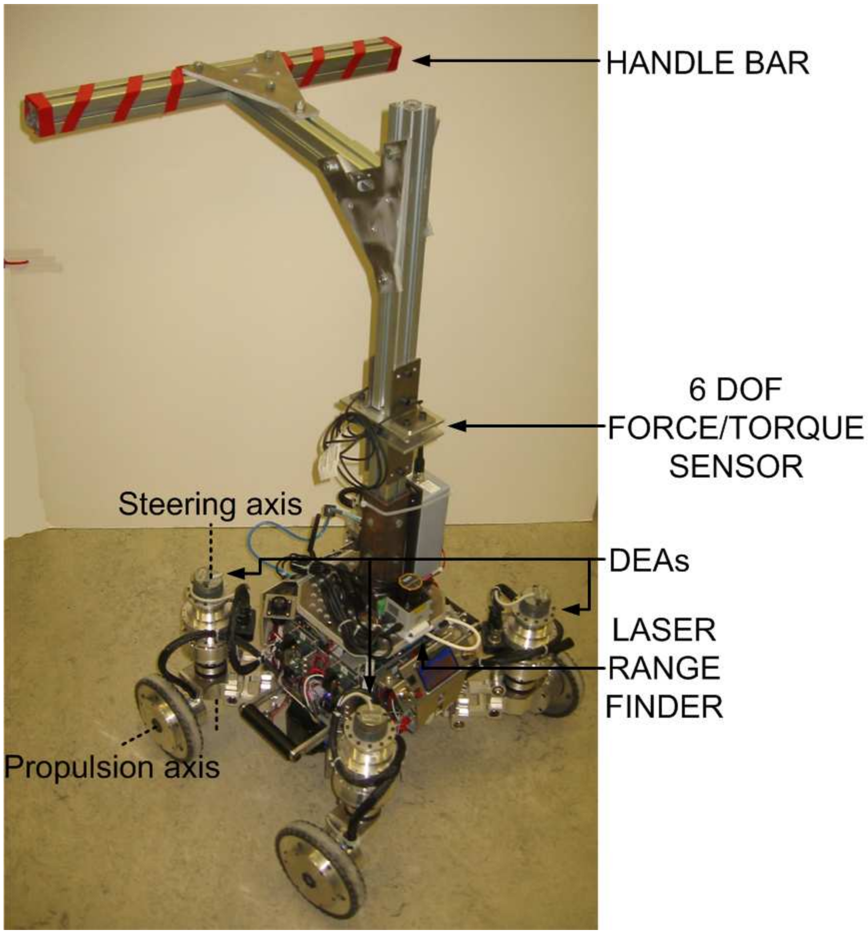

Figure 4.11 Experimental testbed using AZIMUT-3.

Table 4.2 Parameters used in our trials.

Variable	Value	Units
D	1	kg.s^{-2}
K	800	kg.s^{-2}
C_{acc}	5.87e-2	kg.m^{2}
T_{dz}	2e-2	N.m
L_{diag}	0.2501	m
L_{arm}	0.903	m
D_{TTF}	9	kg.s^{-1}
M_{TTF}	10	kg
J_{RTF}	5	kg.m^{2}
D_{RTF}	30	kg.m^{2}.s^{-1}
C_{t_f}	1.5	N
V_{ymax}	0.5	m.s^{-1}
C_{ρ_g}	5	-
C_{γ_g}	3	-
C_{d_goal}	0.8	-
C_{d_path}	0.6	-
C_{speed}	0.1	-

shared-control mechanism. Each of these sets of trials involved seven participants, selected through convenience sampling. They had little or no experience with the platform. Five participants took part in both sets of trials, carried out two weeks apart.

4.6.1 Path Following

The experimental setup has a 5 m path. Each user repeated the test twice in a random order so that in one case, the platform was controlled using F_{APP} and T_{APP}, and in the second F'_{APP} and T'_{APP} was used. The path drawn on the ground provides a general and common context to evaluate our approach, with the objective of demonstrating the feasibility of force-guiding a platform using torque sensors located in steerable wheels. Fig. 4.12 illustrates the average trajectories over all participants (left) and trajectories for two participants using F'_{APP} and T'_{APP} who are representative of the trajectories obtained with these trials. Our intent was not to evaluate the control precision between the platform's path and the desired path, because user's intent repeatability cannot be guaranteed for each participants and between participants. For each trial, we monitored when the force F_{APP} exceeded a minimal threshold at the beginning, and when the velocity reached a minimum at the end of the trial, to have a good estimation of their duration. It took on average 26.5 sec (standard deviation 5.6 sec) for the participants to follow the path using

F_{APP} and T_{APP}, and 23 sec (standard deviation 5.3 sec) with F'_{APP} and T'_{APP}, making both approaches equivalent.

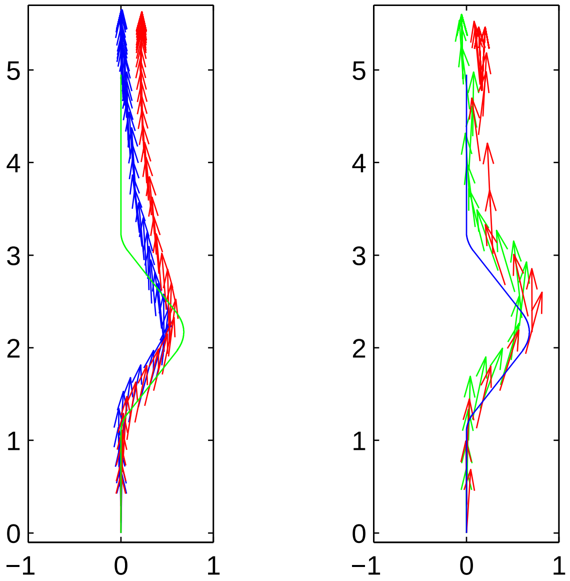

Figure 4.12 Path following observations: (left) Reference trajectory (in green) and average trajectories over all participants using F'_{APP} and T'_{APP} (in blue) and F_{APP} and T_{APP} (in red); (right) reference trajectory (in blue) and trajectories for two participants (User 1 in blue and User 2 in red) force-guiding the platform using F'_{APP} and T'_{APP}.

Fig. 4.13 illustrates F'_{APP}, F_{APP}, T'_{APP} and T_{APP}, for the trajectories of User 1 and User 2. In general, the estimated force and torque derived from the steerable wheels follow similar patterns, with a small delay of 10 ms for F'_{APP} and T'_{APP} which has few or no influence at all on users. This delay is generated by having to derive the user's intent from the torque sensed at each wheel, and can be considered negligible.. For the force, a delay was observed at startup because stiction on one of the wheels led to a inverse torque on the wheel's axis compared to the torque created by the user's push. Disparities were also observed at 11 sec and 20 sec for User 1, and 10 sec and 17 sec for User 2. They can be explained by important lateral forces exerted on the device by users when they come to a turn.

Fig. 4.14 transposes velocities computed with F'_{APP} and T'_{APP} to illustrate the effect of the behavior of the platform according to parameters defined in Section 4.5.3. As expected, the velocities computed are stable and reflect the efforts applied on the platform.

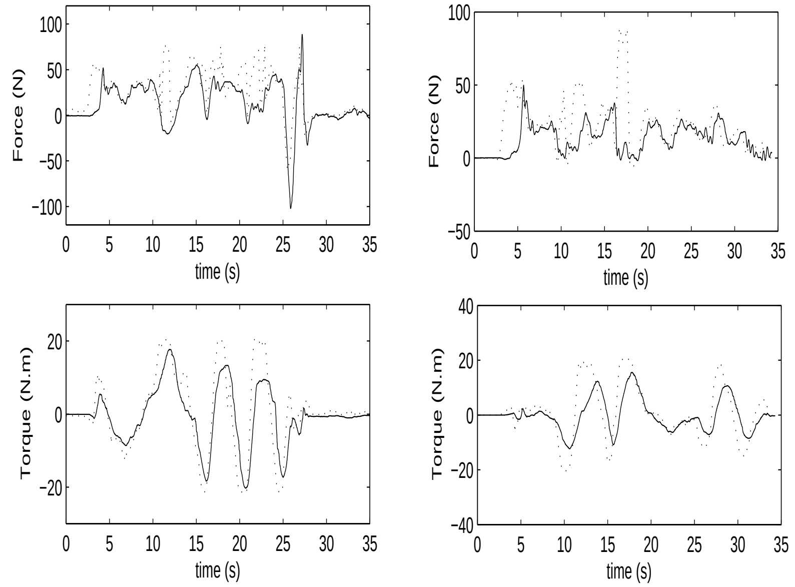

Figure 4.13 F'_{APP} (solid line) and F_{APP} (dotted line) with respect to time (top) and T'_{APP} (solid line) and T_{APP} (dotted line) with respect to time (bottom) for User 1 (left) and User 2 (right).

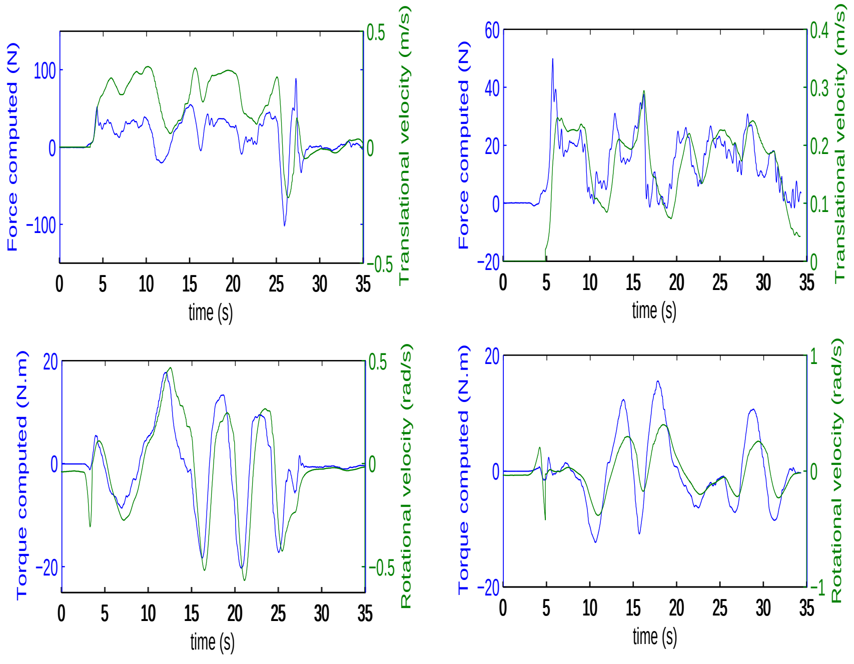

Figure 4.14 F'_{APP} (blue) and V_x (green) with respect to time (top) and T'_{APP} (blue) and V_θ (green) with respect to time for User 1 (left) and User 2 (right).

Finally, Fig. 4.15 presents velocities computed when either F'_{APP} and T'_{APP} or F_{APP} and T_{APP} are used to control the platform, for User 1 and User 2. These velocities are similar with roughly the same small delay, as explained above. Note the response time of the platform can be improved by lowering the values of the mass and inertia factors M_{TTF} and J_{RTF} or the damping factors D_{TTF} and D_{RTF}, but these changes can also impact the stability of the device.

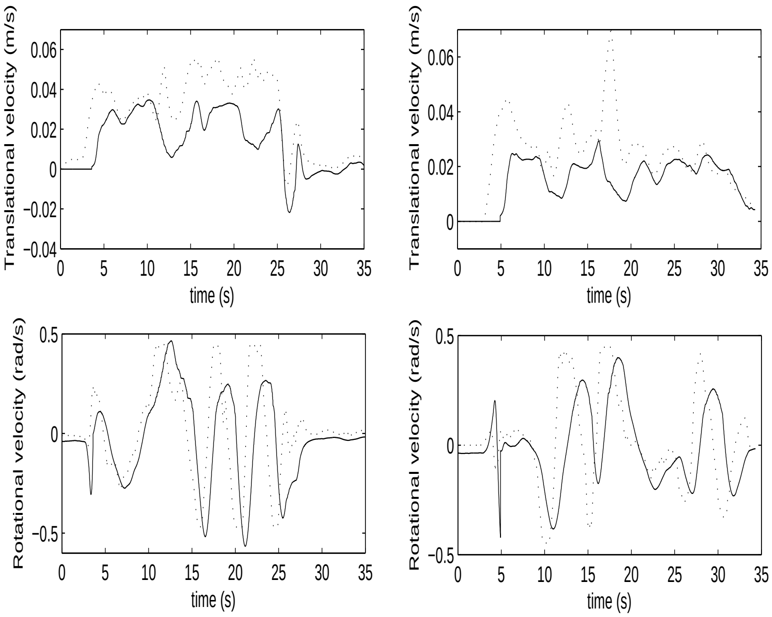

Figure 4.15 Actual V_x computed with F'_{APP} (solid line) and V_x computed with the F_{APP} (dotted line) with respect to time (top) and actual V_θ computed with T'_{APP} (solid line) and V_θ computed with the T_{APP} (dotted line) with respect to time for User 1 (left) and User 2 (right).

4.6.2 Shared-Control Approach

The objective of these trials was to analyze the combined influences of push intent and obstacle avoidance. As shown in Fig. 4.16, participants were blindfolded, and they were told to try to go through the obstacle course. The corridor is 2.64 m wide and 3.68 m long, and three obstacles are randomly disposed at six possible positions. The local 2D occupancy map has a resolution of 10 cm. The platform is in Mode 3 at start-up and goes into Mode 1 once the user starts pushing forward.

Four of the seven participants went through the obstacle course, while the three others stopped in front of obstacles. The latter case happened when the platform was pushed at a slow velocity towards an obstacle and is the expected behavior. Influences of the shared-control mechanism to avoid obstacles are illustrated in Fig. 4.17. The last portion of the trajectory shown right in Fig. 4.17 illustrates the case of pushing the platform directly in front of an obstacle, making it stop. Fig. 4.19 also illustrates a case where the robot is stopped by the wall and an obstacle, assuming that the user wanted to park the platform against the wall.

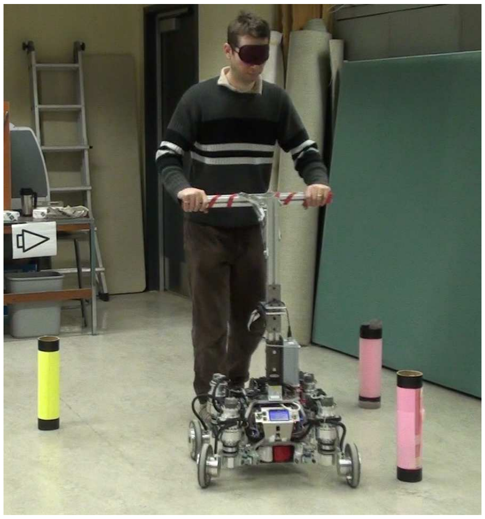

Figure 4.16 Experimental setup for the shared-control mechanism.

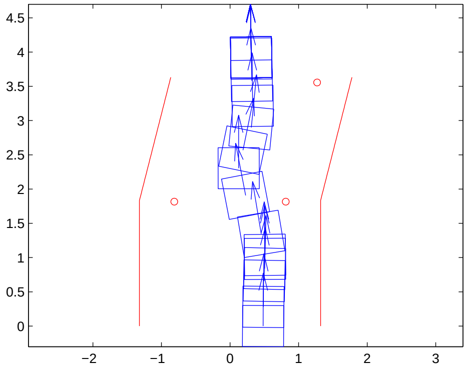

Figure 4.17 Shared-control example with AZIMUT-3 (blue) going through the corridor and avoiding the obstacles (red). Units are in m.

4.7 CONCLUSION AND FUTURE WORK

This paper demonstrates that it is possible to exploit the capabilities of Differential Elastic Actuators for motorization of steerable wheels to make AZIMUT-3 respond softly to forces and torques from a human physically guiding the robot. This approach has the advantage that force-guidance can take place from almost any point on the platform, and reveals to

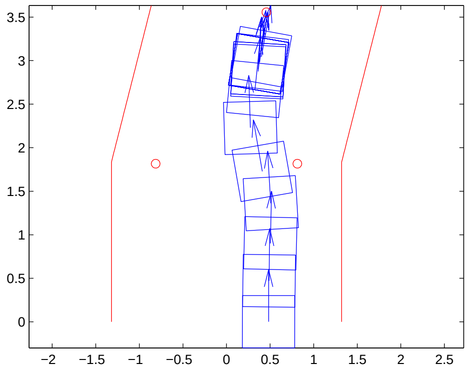

Figure 4.18 Shared-control example with AZIMUT-3 (blue) going through the corridor and avoiding the obstacles (red), but blocked at the end of the experiment. Units are in m.

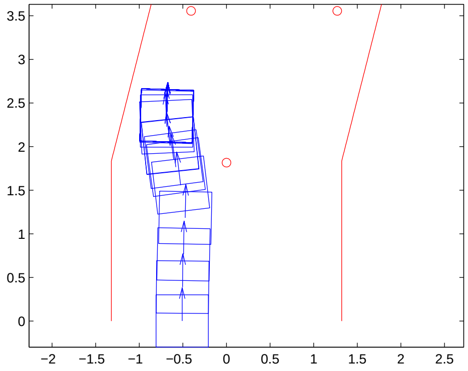

Figure 4.19 Shared-control example with AZIMUT-3 (blue) going through the corridor and avoiding the obstacles (red), but blocked at the end of the experiment. Units are in m.

be a robust and cheap alternative compared to the use of expensive 6 DOF force sensor. The use of elastic actuators for wheel steering, in addition to allow the platform to be force guided, provides an intrisic horizontal suspension to the platform, making it safe in case of collisions. Our shared-control mechanism also demonstrates that user intent can be combined with other sensed modalities of the platform to avoid obstacles while being pushed by a user. In future work, our next implementation will remove some of the constraints on the AZIMUT-3 locomotion modes, and will integrate an inclinometer to be able to move on uneven surfaces.

ACKNOWLEDGEMENTS

The authors would like to thank Dominic Létourneau, Lionel Clavien, François Ferland, Marc-Antoine Legault and all the staff at IntRoLab for making this work possible. They are also grateful to the people who accepted to participate in the trials.

François Michaud holds the Canada Research Chair on Mobile Robotics and Autonomous Intelligent Systems. This project is funded by the Natural Sciences and Engineering Research Council of Canada and the Canadian Foundation for Innovation.

CHAPTER 5

CONCLUSION

This work demonstrates how the omnidirectional platform AZIMUT-3, with its actuated torque-sensing steerable wheels, could be used as a secure, force-guided platform. Using a detailed model to estimate the applied efforts on AZIMUT-3 from torque measurements on its wheels, an algorithm exploits these efforts and obstacle detection using laser range finder data to implement a safe, shared-control approach, without using a global map. Experimental results using the real platform demonstrate feasibility and safe control of the system, with performances similar to using a six degrees of freedom force sensor but at lower cost and with a broader area for shared control. Our implementation also resulted in coupling the simulation environment Webots with the ROS (Robot Operating System) library from Willow Garage, to help develop our approach interfacing the simulation before porting it on AZIMUT-3.

Overall, our work is a first attempt in demonstrating how it is possible to naturally interact by physically moving or positioning a mobile platform in real life settings, a capability which could be useful for instance in the design of powered shopping carts or active walkers. Our shared-control algorithm is also an original contribution compared to what has been presented so far in the research literature. For future work, the approach should be extended to allow ICR not to be restricted to move on lines in each mode, for enhanced motion capability. An inclinometer should also be added to the platform to remove the influence of gravity when the platform operates on an uneven terrain.

ANNEX A

DETAILLED CALCULATION OF APPLIED FORCE AND TORQUE

This appendix presents in more detail calculations to derive Equations (4.16) and (4.17).

A.1 Calculation of F'_{APP}

Starting from Equation (4.8) :

$$\overrightarrow{F_{0\to i}} = -\overrightarrow{F_{GND\to i}} \tag{A.1}$$

which gives using (4.9) :

$$\overrightarrow{0} = \overrightarrow{T_{0\to i}} + \overrightarrow{G_iA_i} \times \overrightarrow{F_{0\to i}} + \overrightarrow{G_iB_i} \times -\overrightarrow{F_{0\to i}} \tag{A.2}$$

Equation (A.2) indicates that there is a simple couple applied on the wheels axes, whose torque M can be expressed as:

$$\overrightarrow{M} = \overrightarrow{A_iB_i} \times \overrightarrow{F_{0\to i}} \tag{A.3}$$

Equation (A.2) can be simplified as :

$$\overrightarrow{T_{0\to i}} = -\overrightarrow{M} \tag{A.4}$$

In frame $\mathcal{F}_{G_i}$, $\overrightarrow{F_{0\to i}}$ can be expressed as:

$$^{\mathcal{F}_{G_i}}\overrightarrow{F_{0\to i}} = \begin{pmatrix} ^{\mathcal{F}_{G_i}}F_{0\to i_x} \\ ^{\mathcal{F}_{G_i}}F_{0\to i_y} \\ 0 \end{pmatrix} \tag{A.5}$$

$$^{\mathcal{F}_{G_i}}\overrightarrow{A_iB_i} = \begin{pmatrix} L_{arm} \\ 0 \\ 0 \end{pmatrix} \tag{A.6}$$

From (A.3), (A.4), (A.5) and (A.6), $^{\mathcal{F}_{G_i}}\overrightarrow{T_{0\to i}}$ can be expressed as:

$$^{\mathcal{F}_{G_i}}\overrightarrow{T_{0\to i}} = \begin{pmatrix} 0 \\ 0 \\ ^{\mathcal{F}_{G_i}}T_{0\to i_z} \end{pmatrix} = -\begin{pmatrix} L_{arm} \\ 0 \\ 0 \end{pmatrix} \times \begin{pmatrix} ^{\mathcal{F}_{G_i}}F_{0\to i_x} \\ ^{\mathcal{F}_{G_i}}F_{0\to i_y} \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -L_{arm}.^{\mathcal{F}_{G_i}}F_{0\to i_y} \end{pmatrix} \tag{A.7}$$

Using the assumption introduced in Section 4.5 (forces collinear with the wheels' propulsion axis are not considered):

$$^{\mathcal{F}_{G_i}}\overrightarrow{F_{0\to i}} = \begin{pmatrix} 0 \\ ^{\mathcal{F}_{G_i}}F_{0\to i_y} \\ 0 \end{pmatrix} \tag{A.8}$$

$\mathcal{F}_{G_i}$ and $\mathcal{F}_P$ are defined in the same plane, thus :

$$^{\mathcal{F}_{G_i}}T_{0\to i_z} =^{\mathcal{F}_P} T_{0\to i_z} = T_{0\to i_z} \tag{A.9}$$

From (A.7) and (A.9) :

$$^{\mathcal{F}_{G_i}}\overrightarrow{F_{0\to i}} = \begin{pmatrix} 0 \\ -\frac{T_{0\to i_z}}{L_{arm}} \\ 0 \end{pmatrix} \tag{A.10}$$

There is a rotation of angle $-\beta_i$ between frames $\mathcal{F}_{G_i}$ and $\mathcal{F}_P$, defining the rotation matrix $R_{\mathcal{F}_{G_i}\mathcal{F}_P}$ matrix :

$$R_{\mathcal{F}_{G_i}\mathcal{F}_P} = \begin{bmatrix} cos(-\beta_i) & -sin(-\beta_i) & 0 \\ sin(-\beta_i) & cos(-\beta_i) & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} cos(\beta_i) & sin(\beta_i) & 0 \\ -sin(\beta_i) & cos(\beta_i) & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{A.11}$$

Using (A.10) and (A.11) :

$$^{\mathcal{F}_P}\overrightarrow{F_{0\to i}} = R_{\mathcal{F}_{G_i}\mathcal{F}_P}.^{\mathcal{F}_{G_i}}\overrightarrow{F_{0\to i}} = \begin{pmatrix} -\frac{T_{0\to i_z}}{L_{arm}}.sin(\beta_i) \\ -\frac{T_{0\to i_z}}{L_{arm}}.cos(\beta_i) \\ 0 \end{pmatrix} \tag{A.12}$$

Now, using Newton's third law :

$$\overrightarrow{F_{0\to i}} = -\overrightarrow{F_{i\to 0}} \tag{A.13}$$

$$\overrightarrow{T_{0\to i}} = -\overrightarrow{T_{i\to 0}} \tag{A.14}$$

Using (A.12) and (A.13), (4.5) can be expressed as :

$$\overrightarrow{F'_{APP}} = \begin{pmatrix} F_{APP_x} \\ F_{APP_y} \\ 0 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{4} -\frac{T_{0\to i_z}}{L_{arm}}.sin(\beta_i) \\ \sum_{i=1}^{4} -\frac{T_{0\to i_z}}{L_{arm}}.cos(\beta_i) \\ 0 \end{pmatrix} \tag{A.15}$$

A.2 Calculation of T'_{APP}

Using (A.3), (A.4) and (4.15), (4.6) can be expressed as :

$$
\begin{aligned}
\overrightarrow{T'_{APP}} &= \sum_{i=1}^{4}\overrightarrow{T_{0\to i}} - \sum_{i=1}^{4}\overrightarrow{PA_i} \times \overrightarrow{F_{i\to 0}} = \sum_{i=1}^{4}\overrightarrow{T_{0\to i}} - \sum_{i=1}^{4}\overrightarrow{PA_i} \times -\overrightarrow{F_{0\to i}} \\
&= \sum_{i=1}^{4}\overrightarrow{T_{0\to i}} - \sum_{i=1}^{4}\left(\begin{pmatrix} L_{diag}.cos({}^{\mathcal{F}_P}\alpha_i) \\ L_{diag}.sin({}^{\mathcal{F}_P}\alpha_i) \\ 0 \end{pmatrix} \times \begin{pmatrix} -\frac{T_{0\to i_z}}{L_{arm}}.sin(\beta_i) \\ -\frac{T_{0\to i_z}}{L_{arm}}.cos(\beta_i) \\ 0 \end{pmatrix}\right) \\
&= \sum_{i=1}^{4}\overrightarrow{T_{0\to i}} \\
&- \sum_{i=1}^{4}\begin{pmatrix} 0 \\ 0 \\ L_{diag}.cos({}^{\mathcal{F}_P}\alpha_i).-\frac{T_{0\to i_z}}{L_{arm}}.cos(\beta_i) - L_{diag}.sin({}^{\mathcal{F}_P}\alpha_i).-\frac{T_{0\to i_z}}{L_{arm}}.sin(\beta_i) \end{pmatrix} \\
&= \sum_{i=1}^{4}\overrightarrow{T_{0\to i}} \\
&- \sum_{i=1}^{4}\begin{pmatrix} 0 \\ 0 \\ -L_{diag}.cos({}^{\mathcal{F}_P}\alpha_i).\frac{T_{0\to i_z}}{L_{arm}}.cos(\beta_i) + L_{diag}.sin({}^{\mathcal{F}_P}\alpha_i).\frac{T_{0\to i_z}}{L_{arm}}.sin(\beta_i) \end{pmatrix} \\
&= \sum_{i=1}^{4}\overrightarrow{T_{0\to i}} - \sum_{i=1}^{4}\begin{pmatrix} 0 \\ 0 \\ -L_{diag}.\frac{T_{0\to i_z}}{L_{arm}}.cos({}^{\mathcal{F}_P}\alpha_i + \beta_i) \end{pmatrix} \qquad \text{(A.16)}
\end{aligned}
$$

ANNEX B
ROS ARCHITECTURE

The ROS framework has been chosen for the development of the project. It offers an execution environment for processes called nodes in ROS.

Data transfers between nodes, which is a recurrent problematic in robotics, is done through topics. Conceptually, topics are queues of messages on which nodes can publish and or subscribe. Messages have a customizable content (integers, floats, arrays, string, ...) and they are identified with their names which are associated to only one topic. It is possible to nest these names in namespaces to reduce this constraint. This is a very versatile mode of communication since neither subscriber or publisher are required for the nodes to function.

ROS also comes with several features to ease monitoring and debugging processes such as plotting tools, simulations engines, and so on.

The ROS architecture developed in the project is as follows. For the sake of simplicity, representations of topics have been reduced to named arrows.

user_force_control. This node is related to what is covered in Sections 4.5.1, 4.5.2 and part of 4.5.4 (for computing user's goal). It receives the state of AZIMUT-3 made of the angular velocities of the propulsion motors, the angular position of the DEAs, and finally the torques perceived on the DEAs' outputs. From this information, a Twist named *user_cmd_vel* is calculated and a global plan is generated. The global plan is made of a goal position and of a succession of positions that lead to it. Both are transmitted to the node *ros_local_avoidance*.

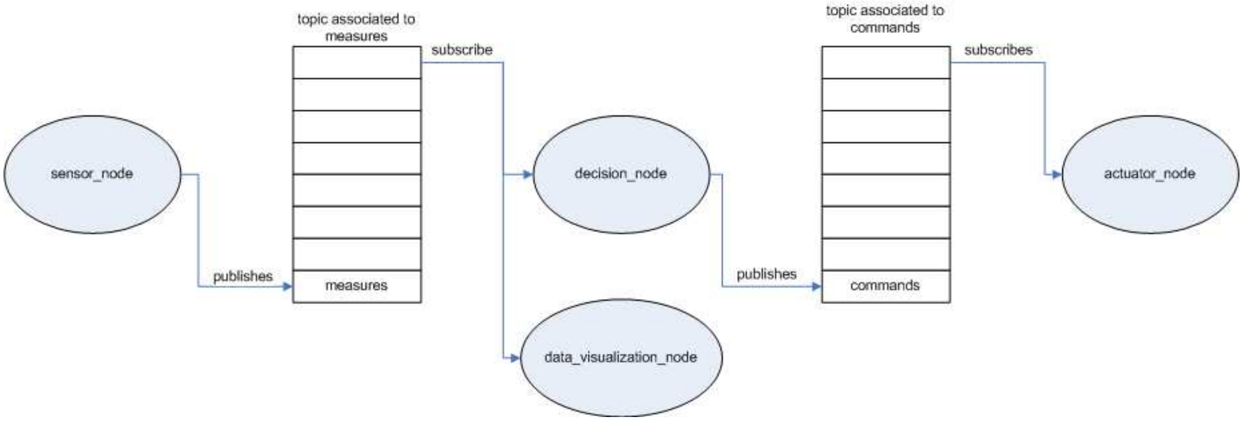

Figure B.1 A classical ROS architecture.

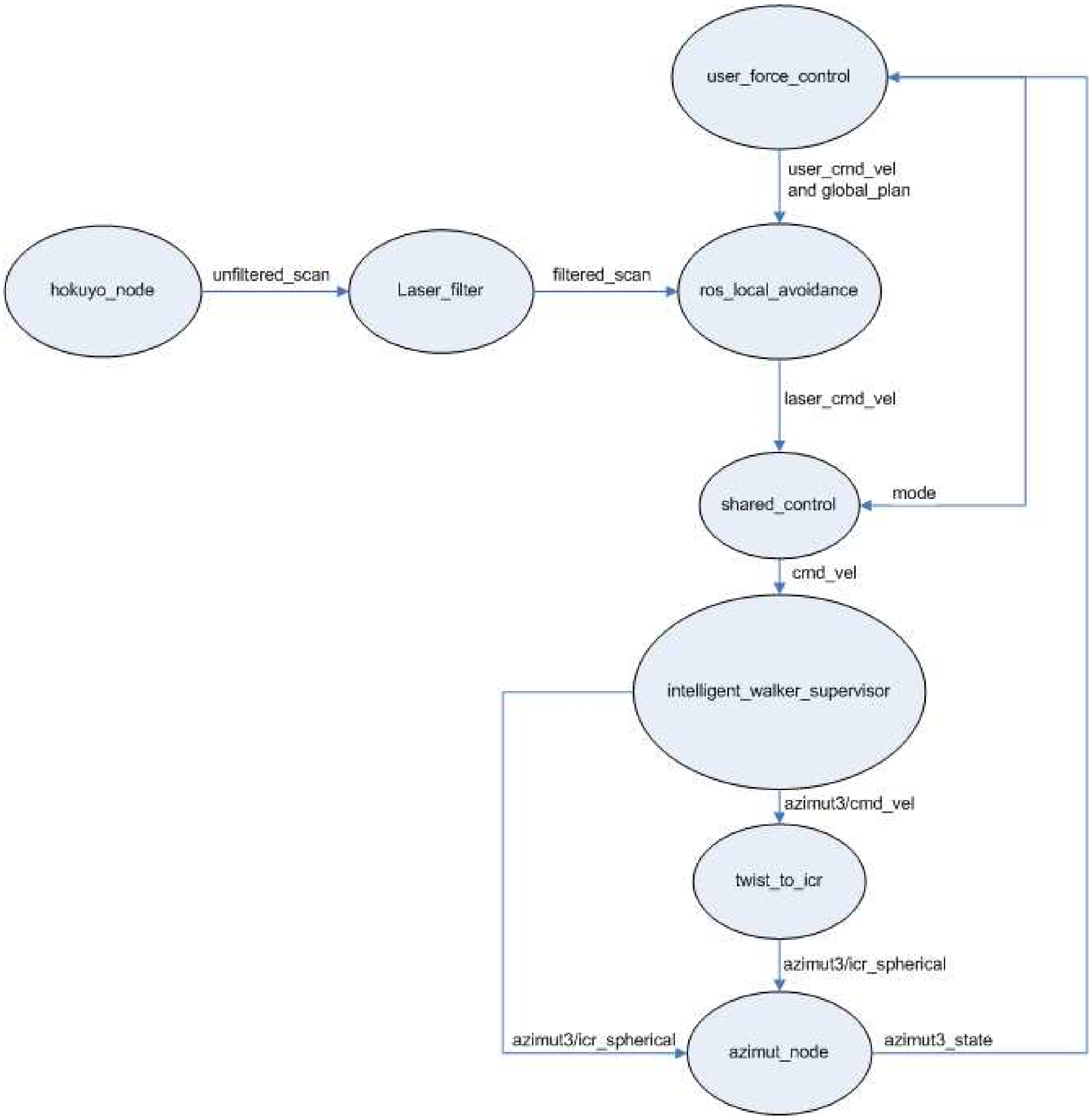

Figure B.2 ROS architecture of AZIMUT-3.

ros_local_avoidance. This node is in charge of what is developed in Section 4.5.4. This node uses the input of the filtered laser scans (provided by nodes *hokuyo_node* for the laser and *laser_filter* for the filter). With these, it internally maintains a costmap that is a discrete and regular representation of its environment that associates a cost to each section defined by the map. The positions of the goal and the positions leading to it are also positioned on that map. The trajectories generated by the DWA algorithm [9] are evaluated with a calculation including distance to the goal and the trajectory to follow, and distance to obstacles to secure the trajectory.

shared_control. This node provides control between the Twist generated by the user and the Twist generated by the autonomous navigation. It also permits to transmit the user Twist directly to the following nodes when the autonomous navigation is not activated, such as when the platform is going backwards in Mode 1, or in the other modes.

intelligent_walker_supervisor. This node compels the ICR to the Mode 1 or 2 when the platform is one of theses modes. It also adjusts the default position of the ICR (and thus the wheels' positions) when a nil Twist is provided, depending on the current mode. Indeed, in most of the cases the mode should stay the same even if a null Twist is given to avoid mode changes. Moreover, we sometimes wants to reset the ICR too, as explained in Section 4.5.2. This is done by this node.

twist_to_icr. This node is in charge of converting a Twist into the ICR defined on a Riemann sphere (see [7]).

azimut_node. This node manages the platform and in particular its low level modules such as the communications on the CAN bus to update the state of the robot which is used by other nodes.

LIST OF REFERENCES

[1] Borenstein, J. and Koren, Y. (September/October 1989) Real-time obstacle avoidance for fact mobile robots. *IEEE Transactions on Systems, Man and Cybernetics*, volume 19, pp. 1179–1187.

[2] Borenstein, J. and Koren, Y. (August 1991) Histogramic in-motion mapping for mobile robot obstacle avoidance. *IEEE Transactions on Robotics and Automation*, volume 7, pp. 535–539.

[3] Chamberland, S., Beaudry, É., Clavien, L., Kabanza, F., Michaud, F. and Lauria, M. (2010) Motion planning for an omnidirectional robot with steering constraints. In *Proceedings IEEE International Conference on Intelligent Robots and Systems.* pp. 4305–4310.

[4] Choset, H. M., Hutchinson, S., Lynch, K. M., Kantor, G., Burgard, W., Kavraki, L. E., Thrun, S. and M, H. (2005) *Principles of Robot Motion: Theory, Algorithms, and Implementation.* MIT Press, 603 pp.

[5] Chuy, J. O., Hirata, Y., Wang, Z. and Kosuge, K. (2005) Motion control algorithms for a new intelligent robotic walker in emulating ambulatory device function. In *Proceedings IEEE International Conference Mechatronics and Automation*, volume 3. pp. 1509–1514.

[6] Chuy, O., Hirata, Y. and Kosuge, K. (2007) Active type robotic mobility aid control based on passive behavior. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems.* pp. 165–170.

[7] Clavien, L., Lauria, M. and Michaud, F. (2010) Instantaneous center of rotation estimation of an omnidirectional mobile robot. In *Proceedings IEEE International Conference on Robotics and Automation.*

[8] Egawa, S., Takeuchi, I., Koseki, A. and Ishii, T. (2004) *Electrically Assisted Walker with Supporter-Embedded Force-Sensing Device.* Springer Berlin / Heidelberg, pp. 313–322.

[9] Fox, D., Burgard, W. and Thrun, S. (1997) The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, volume 4, no. 1, pp. 23–33.

[10] Frémy, J., Ferland, F., Clavien, L., Létourneau, D., Michaud, F. and Lauria, M. (2010) Force-Controlled Motion of a Mobile Platform. In *Proceedings IEEE International Conference on Intelligent Robots and Systems.* pp. 2517–2518.

[11] Frémy, J., Michaud, F. and Lauria, M. (2010) Pushing a robot along-A natural interface for human-robot interaction. In *Proceedings IEEE International Conference on Robotics and Automation.* IEEE, pp. 3440–3445.

[12] Graf, B. and Schraft, R. (2007) Behavior-based path modification for shared control of robotic walking aids. In *Proceedings IEEE International Conference on Rehabilitation Robotics*. pp. 317–322.

[13] Haigh, K. Z. and Yanco, H. (2002) Automation as caregiver: A survey of issues and technologies. In *Proceedings AAAI-02 Workshop "Automation as Caregiver"*. pp. 39–53.

[14] Hirata, Y., Hara, A. and Kosuge, K. (2007) Motion control of passive intelligent walker using servo brakes. *IEEE Transactions on Robotics*, volume 23, no. 5, pp. 981–990.

[15] Hogan, N. (1985) Impedance control: An approach to manipulation: Part ii - dynamics systems measurement control. *Journal of Dynamic Systems, Measurement, and Control*, volume 107, p. 17.

[16] Jiang, K., Seneviratne, L. D. and Earles, S. W. E. (April 1999) A shortest path based path planning algorithm for nonholonomic mobile robots. *Journal of Intelligent and Robotic Systems*, volume 24, pp. 347–366.

[17] Kalman, R. E. (1960) A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, volume 82, pp. 35–45.

[18] Lacey, G. and Rodriguez-Losada, D. (2008) The evolution of Guido. *IEEE Robotics & Automation Magazine*, volume 15, no. 4, pp. 75–83.

[19] Latombe, J.-C. (1991) *Robot Motion Planning*. Springer, 672 pp.

[20] Lauria, M., Legault, M., Lavoie, M. and Michaud, F. (2008) Differential elastic actuator for robotic interaction tasks. In *Proceedings IEEE International Conference on Robotics and Automation*. pp. 3606–3611.

[21] Lauria, M., Nadeau, I., Lepage, P., Morin, Y., Giguere, P., Gagnon, F., Letourneau, D. and Michaud, F. (2006) Kinematical analysis of a four steered wheeled mobile robot. In *Proceedings IEEE International Conference on Industrial Electronics*, volume 4. pp. 3090–3095.

[22] McLachlan, S., Arblaster, J., Liu, O., Miro, J. and Chenoweth, L. (2005) A multi-stage shared control method for an intelligent mobility assistant. In *Proceedings International Conference on Rehabilitation Robotics*. pp. 426–429.

[23] Michaud, F., Letourneau, D., Arsenault, M., Bergeron, Y., Cadrin, R., Gagnon, F., Legault, M., Millette, M., Pare, J., Tremblay, M., Lepage, P., Morin, Y., Bisson, J. and Caron, S. (2005) Multi-modal locomotion robotic platform using leg-track-wheel articulations. *Autonomous Robots*, volume 18, no. 2, pp. 137–156.

[24] Miozzo, M., Morasso, P., Sgorbissa, A. and Zaccaria, R. (2002) Locomaid (the locomotion aid) - A distributed architecture for planning and control. In *Proceedings IEEE International Workshop on Robot and Human Interactive Communication*. pp. 164–169.

[25] Morris, A., Donamukkala, R., Kapuria, A., Steinfeld, A., Matthews, J., Dunbar-Jacob, J. and Thrun, S. (2003) A robotic walker that provides guidance. In *Proceedings IEEE International Conference on Robotics and Automation*, volume 1. pp. 25–30.

[26] Park, T.-J., Han, C.-S., Han, J.-S. and Lee, H.-G. (2004) Development of the gait assistant mobile robot using ergonomic design. In *Proceedings IEEE International Conference on Robotics and Automation*, volume 2. pp. 2037–2042.

[27] Prassler, E., Lawitzky, G., Stopp, A. and Erwin (2004) *Advances in Human-Robot Interaction*. Springer, 414 pp.

[28] Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R. and Ng, A. (2009) Ros: an open-source robot operating system. In *Open-Source Software Workshop, IEEE International Conference on Robotics and Automation*.

[29] Research, A. (2010) *Personal Robotics 2010*. Technical report, ABI Research. URL `http://www.abiresearch.com/research/1003675`.

[30] Robinson, D. (2000) *Design and Analysis of Series Elasticity in Closed Loop Actuator Force Control*. Ph.D. thesis, Massachussets Institute of Technology, Cambridge, Mass., USA.

[31] Rodriguez-Losada, D., Matia, F., Jimenez, A., Galan, R. and Lacey, G. (2005) Implementing map based navigation in guido, the robotic smartwalker. In *Proceedings IEEE International Conference on Robotics and Automation*. pp. 3390–3395.

[32] Sabatini, A., Genovese, V. and Pacchierotti, E. (2002) A mobility aid for the support to walking and object transportation of people with motor impairments. In *IEEE International Conference on Intelligent Robots and System*, volume 2. pp. 1349–1354.

[33] Shi, Y., Kotani, S. and Mori, H. (2002) A pre-journey system of the robotic travel aid. In *Proceedings IEEE International Conference on Robotics and Automation*, volume 4. pp. 4149–4154.

[34] Shim, H.-M., Lee, E.-H., Shim, J.-H., Lee, S.-M. and Hong, S.-H. (2005) Implementation of an intelligent walking assistant robot for the elderly in outdoor environment. In *Proceedings International Conference on Rehabilitation Robotics*. pp. 452–455.

[35] Siegwart, D. R. and Nourbakhsh, I. R. (2004) *Introduction to Autonomous Mobile Robots*. MIT Press, 321 pp.

[36] Simpson, R. C. and Rentschler, A. (2006) Intelligent mobility aids. *Wiley Encyclopedia of Biomedical Engineering*.

[37] Spenko, M., Yu, H. and Dubowsky, S. (2006) Robotic personal aids for mobility and monitoring for the elderly. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, volume 14, no. 3, pp. 344–351.

[38] Thrun, S. (May 2001) A probabilistic on-line mapping algorithm for teams of mobile robots. *The International Journal of Robotics Research*, volume 20, no. 5, pp. 335–363.

[39] Thrun, S., Fox, D., Burgard, W. and Dellaert, F. (May 2001) Robust monte carlo localization for mobile robots. *Artificial Intelligence*, volume 128, no. 1-2, pp. 99–141.

[40] Wasson, G., Sheth, P., Alwan, M., Granata, K., Ledoux, A. and Huang, C. (2003) User intent in a shared control framework for pedestrian mobility aids. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3. pp. 2962–2967.

[41] Wasson, G., Sheth, P., Huang, C. and Alwan, M. (2008) *Intelligent Mobility Aids for the Elderly*. Humana Press, pp. 53–76.

[42] Williamson, M. (1995) *Series Elastic Actuators*. Master's thesis, Massachussets Institute of Technology, Cambridge, Mass., USA.